Volume 10 Numbers 1—2

ACTA CYBERNETICA

Szeged, 1991

Information for authors: Acta Cybernetica publishes only original papers in English in the field of computer sciences. Review papers are accepted only exceptionally. Manuscripts should be sent in triplicate to one of the Editors. The manuscript must be typed double-spaced on one side of the paper only. For the form of references, see one of the articles previously published in the journal.

Acta Cybernetica, Tom. 10, Fasc. 1-2, Szeged, 1991

Analysis of the SCAN service polling model

Brian D. Bunday*

Abstract

A performance analysis is given for a polling system in which the server polls N stations back and forth according to the so-called SCAN system. Messages arrive at each station in Poisson fashion at an average rate λ. The number of characters in a message has a geometric distribution with mean $1/\sigma$. The service time per character, b, and the switchover time between adjacent stations, r, are both assumed to be constant.

An exact analysis is given but because of associated computational problems this has limited application. Thus a second approximate analysis which allows systems with a large number of stations to be treated, has been developed. In both cases it is a simple matter to calculate such performance measures as average polling cycle time and mean response time at each station.

Keywords: Machine interference models, Markov chain, Performance evaluation, Polling models, SCAN systêm.

1 Introduction

We consider a SCAN polling model in which the server polls N stations in the order $1, 2, 3, \ldots, N-1, N, N-1, N-2, \ldots, 3, 2, 1, 2, 3$, etc. Messages arrive at each station in Poisson fashion at an average rate λ and the number of characters in a message has a geometric distribution with mean $1/\sigma$. Thus at the completion of each character service the message completes its service with probability σ, and does not complete service with probability $1-\sigma$. Yet another viewpoint is that we have single buffer stations, and are modelling error-prone transmission channels. Here σ is the probability that the message departs (successful transmission). The service time per character, b, and the switchover time between adjacent stations, r, are both assumed to be constants.

Polling systems have been the subject of much study for a number of years. A good review and bibliography is given by Takagi [8,10] who also mentions the SCAN system as well as the geometric distribution of characters referred to as a round-robin system in Takagi [9]. For an up-to-date bibliography see Takagi [11]. Coffman and Gilbert [4] consider a continouos polling model of SCAN service. Other analyses of this system can be found in the work of Coffman and Hofri [5], Swartz [7] and Takagi and Murata [12]. In fact the SCAN system has also been considered in the context of machine interference models by Bunday and Mack [3] and Bunday, El-Badri and Supanekar [1]. Similar mathematical models have also

*University of Bradford, U.K.

been treated by Kim and Koenigsberg [6] in their analysis of automatic loading and retrieval on carousel conveyor systems.

2 Exact Analysis

The assumption of Poisson arrival of messages at average rate λ means that if a station is empty at time t, then the chance that it is still empty after a further period of time T is

$$\exp(-\lambda T) \tag{1}$$

whereas the chance that a message has arrived by this time is

$$1 - \exp(\lambda T) \tag{2}$$

independent of T.

We denote by

$$\{u_1/u_2u_3 \ldots u_N\}$$

the left to right traverse of the stations (i.e. in the order $1, 2, \ldots, N-1, N$) in which the server *leaves* station 1 in state u_1 and *finds* station i ($i = 2, \ldots, N$) in state u_i, where $u_i = 0$ denotes no message at station i, whereas $u_i = 1$ indicates that there is a message at station i.

$$p\{u_1/u_2u_3 \ldots u_N\}$$

is the probability that the server encounters this situation on a left to right traverse.

Similarly right to left traverses of the stations, when they are polled in the order $N, N-1, \ldots, 3, 2, 1$, occur with probability (in an obvious notation)

$$p\{v_1v_2 \ldots v_{N-1}/v_N\}$$

and because of the symmetry of the system,

$$p\{v_1v_2 \ldots v_{N-1}/v_N\} = p\{v_N/v_{N-1} \ldots v_2v_1\}. \tag{3}$$

The 2^N probabilities $p\{u_1/u_2 \ldots u_N\}$ satisfy the condition

$$\sum_{\{u_1/u_2 \ldots u_N\}} p\{u_1/u_2 \ldots u_N\} = 1, \tag{4}$$

along with the usual equilibrium equation for the finite Markov chain, viz.

$$p\{u_1/u_2 \ldots u_N\} = \sum_{\{v_1v_2 \ldots v_{N-1}/v_N\}} p\{v_1v_2 \ldots v_{N-1}/v_N\} P\{u_1/u_2 \ldots u_N | v_1v_2 \ldots v_{N-1}/v_N\}, \tag{5}$$

which on account of (3) can be written

$$p\{u_1/u_2 \dots u_N\} = \sum_{\{v_1/v_2 \dots v_N\}} p\{v_1/v_2 \dots v_N\} P\{u_1/u_2 \dots u_N | v_N v_{N-1} \dots v_2/v_1\}, \tag{6}$$

for each $\{u_1/u_2 \dots u_N\}$.

Here $P\{u_1/u_2 \dots u_N | v_N v_{N-1} \dots v_2/v_1\}$ is the conditional probability of encountering the situation described by $\{u_1/u_2 \dots u_N\}$ on the left to right traverse following the right to left traverse described by $\{v_N v_{N-1} \dots v_2/v_1\}$.

The computation of this transition probability proceeds along the following lines. For the jth station, the time that elapses between it being left on a right to left traverse and next visited on the following left to right traverse, is just $2r + bv_N$ for $j = 2$, and

$$\tau_j = 2(j-1)r + b\left[\sum_{i=2}^{j-1} u_i + \sum_{i=N-j+2}^{N} v_i\right] \tag{7}$$

for $j = 3, 4, \dots, N$.

If we denote by $Pr_j(u_j | v_{N-j+1})$ the probability that the server finds station j in state u_j on a left to right traverse, having found it in state v_{N-j+1} on the previous right to left traverse, then from (1) and (2), for $j = 2, 3, \dots N-1$,

$$Pr_j(0|0) = \exp(-\lambda\tau_j), \qquad Pr_j(1|0) = 1 - \exp(-\lambda\tau_j)$$

$$Pr_j(0|1) = \sigma\exp(-\lambda\tau_j), \qquad Pr_j(1|1) = 1 - \sigma\exp(-\lambda\tau_j). \tag{8}$$

For $j = 1$ we need the probability of *leaving* station 1 in state u_1 having just *found* it in state v_N. Thus

$$Pr_1(0|0) = 1 \quad Pr_1(1|0) = 0, \quad Pr_1(0|1) = \sigma, \quad Pr_1(1|1) = 1 - \sigma \tag{9}$$

For $j = N$ we consider the probability of *finding* station N in state u_N having *left* it in state v_1. Thus

$$Pr_N(0|0) = \exp(-\lambda\tau_N), \qquad Pr_N(1|0) = 1 - \exp(\lambda\tau_N)$$

$$Pr_N(0|1) = 0, \qquad Pr_N(1|1) = 1. \tag{10}$$

There is some abuse of notation here since $Pr_j(u_j | v_{N-j+1})$ also depends on the other u's and v's through τ_j. We use (7), (8), (9) and (10) to compute the transition probability as

$$P\{u_1/u_2 \dots u_N | v_N \dots v_2/v_1\} = \Pi_{j=1}^{N} Pr_j(u_j | v_{N-j+1}). \tag{11}$$

We use (4) and (6) to obtain 2^N independent equations for the 2^N state probabilities. This analysis can be used in conjunction with modern computing facilities to obtain exact solutions for $N \leq 10$.

It is perhaps worth mentioning that provided we have symmetry in the "line of stations" so that (3) holds then the same analysis will hold for the inhomogeneous system with message arrival rate λ_j and character service time b_j at station j. We simply replace λ by the appropriate λ_j in (8) and (10) whilst (7) is replaced by

$$r_j = 2(j-1)r + \sum_{i=2}^{j-1} b_i u_i + \sum_{i=N-j+2}^{N} b_{N-i+1} v_i.$$

For further details and applications to machine interference models the reader is referred to the report by Bunday and Khorram (2).

3 Approximate Analysis

The modelling in the previous section calls for the solution of 2^N simultaneous linear equations. In the present state of computing this limits the method to values of $N \leq 10$.

A traverse of the stations begins when the server leaves polling station $1(N)$ and ends when he leaves station $N(1)$. Thus at most $N-1$ characters are serviced on a traverse. A complete cycle consists of two consecutive traverses.

In the equilibrium state the probability that station m is message free when polled from the left is denoted by

$$p_m \quad (2 \leq m \leq N) \tag{12}$$

The probability that station m is message free when polled from the right is denoted by p'_m and from the symmetry

$$p'_m = p_{N+1-m} \quad (m = N-1, N-2, \ldots, 2, 1). \tag{13}$$

$$p_m = \Sigma^{(m)} p\{u_1/u_2 \ldots u_N\} \tag{14}$$

where $\Sigma^{(m)}$ denotes summation over those states for which $u_m = 0$.

For $m \geq 2$ the left partial cycle time (L.P.C.T.) of station m is defined to be the time that elapses between the server's departure from station m and subsequent arrival at this station, the journey being made via station 1. It does not include the service time, if any, for station m, on either occasion. The right partial cycle time (R.P.C.T) in which the journey is made via station N is similarly defined. In the equilibrium state the L.P.C.T. of station m and the R.P.C.T. of station $N+1-m$ have identical distributions.

We let

$$u(l,m) \quad (2 \leq m \leq N, 0 \leq l \leq 2m-3) \tag{15}$$

denote the probability that in the L.P.C.T. of station m, l characters are serviced (l of the stations have messages). Thus $u(l,m)$ is the equilibrium probability that the L.P.C.T. of station m takes the value

$$q(l,m) = 2(m-1)r + bl. \tag{16}$$

Similarly

$$v(l,m) \tag{17}$$

is the probability that the L.P.C.T. of station m takes the value $q(l,m)$ given that station m had no message, when it was approached from the right just prior to the left partial cycle, whereas

$$w(l,m) \tag{18}$$

is the probability that the L.P.C.T. of station m takes the value $q(l,m)$ given that station m had a message just before the partial cycle.

Thus

$$u(l,m) = p'_m v(l,m) + (1-p'_m)w(l,m)$$

i.e.

$$u(l,m) = p_{N+1-m}v(l,m) + (1-p_{N+1-m})w(l,m) \tag{19}$$

on using (13).

For station 2, $u(0,2) = p'_1$ and $u(1,2) = 1 - p'_1$ so that

$$u(0,2) = p_N \text{ and } u(1,2) = 1 - p_N. \tag{20}$$

On using (1) we obtain

$$\begin{aligned} p_2 \;\; = \;\; & p'_2 v(0,2) \exp\left[-\lambda q(0,2)\right] + (1-p'_2)\sigma w(0,2) \exp\left[-\lambda q(0,2)\right] \\ & + p'_2 v(1,2) \exp\left[-\lambda q(1,2)\right] + (1-p'_2)\sigma w(1,2) \exp\left[-\lambda q(1,2)\right], \end{aligned}$$

i.e.,

$$\begin{aligned} p_2 \;\; = \;\; & p_{N-1}\{v(0,2)\exp[-\lambda q(0,2)] + v(1,2)\exp[-\lambda q(1,2)]\} \\ & + (1-p_{N-1})\sigma\{w(0,2)\exp-[\lambda q(0,2)] + w(1,2)\exp[-\lambda q(1,2)]\} \end{aligned} \tag{21}$$

For $2 < m \leq N$ it is possible to relate the distribution of the L.P.C.T. for station m to that of station $(m-1)$. On using (1), (2) and (13) we obtain for $2 < m \leq N$

$$\begin{aligned} u(0,m) \;\; &= \;\; p_{N+2-m}v(0,m-1)\exp[-\lambda q(0,m-1)] \\ u(1,m) \;\; &= \;\; p_{N+2-m}v(1,m-1)\exp[-\lambda q(1,m-1)] \\ & \quad + p_{N+2-m}v(0,m-1)\{1-\exp[-\lambda q(0,m-1)]\} \\ & \quad + (1-p_{N+2-m})\sigma w(0,m-1)\exp[-\lambda q(0,m-1)] \end{aligned}$$

and for $1 < l < 2m-4$

$$\begin{aligned} u(l,m) \;\; = \;\; & p_{N+2-m}v(l,m-1)\exp[-\lambda q(l,m-1)] \\ & + p_{N+2-m}v(l-1,m-1)\{1-\exp[-\lambda q(l-1,m-1)]\} \\ & + (1-p_{N+2-m})\sigma w(l-1,m-1)\exp[-\lambda q(l-1,m-1)] \\ & + (1-p_{N+2-m})\sigma w(l-2,m-1)\{1-\exp[-\lambda q(l-2,m-1)]\} \\ & + (1-p_{N+2-m})(1-\sigma)w(l-2,m-1) \end{aligned} \tag{22}$$

$$\begin{aligned} u(2m-4,m) \;\; = \;\; & p_{N+2-m}v(2m-5,m-1)\{1-\exp[-\lambda q(2m-5,m-1)]\} \\ & + (1-p_{N+2-m})\sigma w(2m-5,m-1)\exp[-\lambda q(2m-5,m-1)] \\ & + (1-p_{N+2-m})\sigma w(2m-6,m-1) \\ & \times\{1-\exp[-\lambda q(2m-6,m-1)]\} \\ & + (1-p_{N+2-m})(1-\sigma)w(2m-6,m-1) \\ u(2m-3,m) \;\; = \;\; & (1-p_{N+2-m})\sigma w(2m-5,m-1) \\ & \times\{1-\exp[-\lambda q(2m-5,m-1)]\} \\ & + (1-p_{N+2-m})(1-\sigma)w(2m-5,m-1) \end{aligned}$$

For station m, (21) generalises to

$$\begin{aligned} p_m &= p_{N+1-m} \sum_{l=0}^{2m-3} v(l,m) \exp[-\lambda q(l,m)] \\ &\quad + (1-p_{N+1-m})\sigma \sum_{l=0}^{2m-3} w(l,m) \exp[-\lambda q(l,m)]. \end{aligned} \tag{23}$$

Now for each m the probabilities $u(l,m)$ form a complete distribution so that

$$\sum_{l=0}^{2m-3} u(l,m) = 1. \tag{24}$$

For $m = 2$ the result is clearly true from (20) and using (19) and (22) and some elementary but messy algebra the general result is easily established by induction.

The exact relationship between $v(l,m)$ and $w(l,m)$ is enormously complicated and involves the probabilities $p\{u_1/u_2 \dots u_N\}$. To bring these latter quantities into consideration makes the problem impossible from a computational aspect if N is large. However if N is large and/or λb small the conditioning effect of the state of one station will become insignificant. In this situation (19) shows that

$$u(l,m) = v(l,m) = w(l,m). \tag{25}$$

Thus for $2 < m \le N$, (22) takes the form,

$$\begin{aligned} u(0,m) &= p_{N+2-m}u(0,m-1) \exp[-\lambda q(0,m-1)] \\ u(1,m) &= p_{N+2-m}u(1,m-1) \exp[-\lambda q(1,m-1)] \\ &\quad + p_{N+2-m}u(0,m-1)\{1-\exp[-\lambda q(0,m-1)]\} \\ &\quad + (1-p_{N+2-m})\sigma u(0,m-1) \exp[-\lambda q(0,m-1)] \end{aligned}$$

and for $1 < l < 2m-4$,

$$\begin{aligned} u(l,m) &= p_{N+2-m}u(l,m-1) \exp[-\lambda q(l,m-1)] \\ &\quad + p_{N+2-m}u(l-1,m-1)\{1-\exp[-\lambda q(l-1,m-1)]\} \\ &\quad + (1-p_{N+2-m})\sigma u(l-1,m-1) \exp[-\lambda q(l-1,m-1)] \\ &\quad + (1-p_{N+2-m})u(l-2,m-1)\{1-\sigma \exp[-\lambda q(l-2,m-1)]\} \end{aligned} \tag{26}$$

$$\begin{aligned} u(2m-4,m) &= p_{N+2-m}u(2m-5,m-1)\{1-\exp[-\lambda q(2m-5,m-1)]\} \\ &\quad + (1-p_{N+2-m})\sigma u(2m-5,m-1) \exp[-\lambda q(2m-5,m-1)] \\ &\quad + (1-p_{N+2-m})u(2m-6,m-1) \\ &\quad \times\{1-\sigma \exp[-\lambda q(2m-6,m-1)]\} \\ u(2m-3,m) &= (1-p_{N+2-m})u(2m-5,m-1) \\ &\quad \times\{1-\sigma \exp[-\lambda q(2m-5,m-1)]\}. \end{aligned}$$

The *extended* version of (23) becomes

$$p_1 = p_N + \sigma(1-p_N)$$

$$p_m = (p_{N+1-m} + \sigma - \sigma p_{N+1-m}) \sum_{l=0}^{2m-3} u(l,m) \exp[-\lambda q(l,m)], \quad 1 < m < N, \quad (27)$$

$$p_N = p_1 \sum_{l=0}^{2m-3} u(l,N) \exp[-\lambda q(l,N)].$$

Here p_1 has been defined as the probability that station 1 is *left* without a message on a left to right traverse.

It is possible to solve (20), (26) and (27) numerically. Just $N^2 - 1$ unknown quantities are involved. From an initial approximation for $p_1, p_2, \ldots, p_N$ (derived from the solution for the previous lower value of N) it is possible to use the equations recursively to compute a better approximation. In this way the p_m were calculated to an accuracy of 10^{-6}.

Of course for values of $N \leq 10$ the exact and approximate analysis can be compared. In neither case are the required computations trivial, but as expected the calculations indicated that the approximate analysis is exact for $N = 2$, is at its worst for N around 6, 7 and 8, but improves again at $N = 9$ and 10. One might add that in all situations the errors involved were quite small, the more so for the performance measures considered in the next section. The relative values of λb and λr had an effect on the validity of the approximate analysis.

Indeed when $b = 0$ the approximate analysis is again exact. Any delays in this case are due to switchover times in order to reach stations with messages and to leaving stations with messages following unsuccessful transmission. In this case (27) takes the form

$$p_1 = p_N + \sigma(1 - p_N)$$

$$p_m = c^{m-1}(p_{N-m+1} + \sigma - \sigma p_{N-m+1}), \quad 2 \leq m \leq N-1, \quad (28)$$

$$p_N = c^{N-1} p_1$$

where $c = \exp(-2\lambda r)$. These equations are easily solved in pairs. It is perhaps worth commenting that this particular solution is not immediately apparent from the exact analysis of Section 2.

4 Performance Measures

The mean time τ for the server to traverse the stations in one direction is given by

$$\tau = (N-1)r + b \sum_{\{u_1/u_2 \ldots u_N\}} p\{u_1/u_2 \ldots u_N\} S\{u_1/u_2 \ldots u_N\} \quad (29)$$

where

$$S\{u_1/u_2 \ldots u_N\} = \sum_{i=2}^{N} u_i \quad (30)$$

is the number of character services carried out on the traverse $\{u_1/u_2 \dots u_N\}$. The mean time for a complete cycle is just 2τ.

If we use (14) a second more useful form for τ is

$$\tau = (N-1)r + b\sum_{j=2}^{N}(1-p_j). \tag{31}$$

The mean time that station m is free of messages in its L.P.C.T. is

$$\begin{aligned} L_m &= p_{N+1-m}\sum_{l=0}^{2m-3} v(l,m)\left\{\int_0^{q(l,m)} \lambda t e^{-\lambda t}dt + q(l,m)\exp[-\lambda q(l,m)]\right\} \\ &\quad +\sigma(1-p_{N+1-m})\sum_{l=0}^{2m-3} w(l,m)\left\{\int_0^{q(l,m)} \lambda t e^{-\lambda t}dt + q(l,m)\exp[-\lambda q(l,m)]\right\} \end{aligned}$$

Thus, on using (23), for $2 \le m \le N-1$ we have

$$L_m = \frac{\sigma + (1-\sigma)p_{N+1-m} - p_m}{\lambda}. \tag{32}$$

If F_m denotes the mean time that station m is free of messages in a complete cycle

$$F_m = L_m + L_{N+1-m} = \frac{\sigma[1 - p_m + 1 - p_{N+1-m}]}{\lambda}. \tag{33}$$

for $2 \le m \le N-1$
while

$$F_1 = F_N = \frac{\sigma(1-p_N)}{\lambda}. \tag{34}$$

Thus if β_m is the proportion of time that station m is free of messages (not blocked)

$$\beta_m = F_m/2\tau. \tag{35}$$

The major performance measure at station m is the mean message response time, which is the average time that an arbitrary message, arriving at the station, takes from its arrival to its service completion. We denote this by $\mathrm{E}[T_m]$. Other performance measures such as the mean waiting time $\mathrm{E}[W_m]$ and throughput γ_m are easily calculated from $\mathrm{E}[T_m]$

$$\mathrm{E}[T_m] = \mathrm{E}[W_m] + b, \tag{36}$$

$$\gamma_m = 1/\left[\mathrm{E}[T_m] + \frac{1}{\lambda}\right]. \tag{37}$$

But $\mathrm{E}[T_m]$ can be calculated from the result

$$\beta_m = \frac{1}{\lambda}/\left[\frac{1}{\lambda} + \mathrm{E}[T_m]\right]$$

whence

$$\lambda \mathrm{E}[T_m] = \frac{1}{\beta_m} - 1. \tag{38}$$

The case of zero switchover times gives a system identical, insofar as the mean response time is concerned, to the M/D/1/N queue with first-come, first served queue discipline. In that case

$$\lambda \mathrm{E}[T] = \lambda N b - 1 + 1 \Big/ \left[1 + \sum_{n=1}^{N-1} \begin{bmatrix} N-1 \\ n \end{bmatrix} \prod_{j=1}^{n} (e^{\lambda j b} - 1) \right]. \tag{39}$$

5 Some Numerical Results

The problem has five parameters N, σ, λ, b and r and in consequence coverage of a wide range of values would make extensive demands on space in any tables. Suffice it to say that the computer program which did the calculations easily deals with other values of the above parameters.

Table 1 gives details of the performance measures at individual stations and illustrates the inhomogeneity of the performance along the line.

In Table 1 the column headed $L\beta_m$ records the quantity $L_m/2\tau$. From (35) β_m represents the proportion of time that station m is not blocked and since of course

$$\beta_m = L\beta_m + L\beta_{N+1-m} \tag{40}$$

for $2 \le m \le N-1$, $L\beta_m$ indicates how this proportion is divided between the left partial cycles and the right partial cycles.

Table 2 details the system performance. The proportion of time on average that stations in the system are not blocked is

$$\beta = \frac{\sum_{m=1}^{N} \beta_m}{N} \tag{41}$$

The average time that a message at a station takes from its arrival to service completion is given by

$$\lambda \mathrm{E}[T] = \frac{1}{\beta} - 1. \tag{42}$$

Table 1

Performance Measures at Individual Stations

$\sigma = 0.1$

N	λb	$N\lambda r$	m	$L\beta_m$	β_m	$\lambda E[T_m]$	p_m
4	0.01	0.01	2	0.2738	0.8251	0.2120	0.9110
			3	0.5513	0.8251	0.2120	0.9078
			4	0.7583	0.7583	0.3187	0.8335
7	0.01	0.01	2	0.1247	0.7703	0.2982	0.8672
			3	0.2550	0.7704	0.2980	0.8647
			4	0.3852	0.7704	0.2980	0.8623
			5	0.5154	0.7704	0.2980	0.8598
			6	0.6456	0.7703	0.2982	0.8574
			7	0.6770	0.6770	0.4771	0.7579
12	0.01	0.01	2	0.0534	0.6755	0.4804	0.7273
			3	0.1167	0.6757	0.4799	0.7244
			4	0.1799	0.6759	0.4796	0.7215
			5	0.2432	0.6760	0.4794	0.7186
			6	0.3064	0.6760	0.4792	0.7158
			7	0.3696	0.6760	0.4792	0.7130
			8	0.4328	0.6760	0.4794	0.7102
			9	0.4959	0.6759	0.4796	0.7074
			10	0.5590	0.6757	0.4799	0.7047
			11	0.6221	0.6755	0.4804	0.7020
			12	0.5361	0.5361	0.8655	0.5471
18	0.01	0.01	2	0.0229	0.5157	0.9392	0.5474
			3	0.0544	0.5159	0.9383	0.5441
			4	0.0858	0.5161	0.9375	0.5409
			5	0.1172	0.5163	0.9369	0.5377
			6	0.1486	0.5164	0.9364	0.5345
			7	0.1800	0.5165	0.9360	0.5314
			8	0.2113	0.5166	0.9357	0.5282
			9	0.2427	0.5166	0.9356	0.5252
			10	0.2740	0.5166	0.9356	0.5221
			11	0.3053	0.5166	0.9357	0.5191
			12	0.3365	0.5165	0.9360	0.5162
			13	0.3678	0.5164	0.9364	0.5132
			14	0.3990	0.5163	0.9369	0.5103
			15	0.4303	0.5161	0.9375	0.5074
			16	0.4615	0.5159	0.9383	0.5046
			17	0.4927	0.5157	0.9392	0.5018
			18	0.3576	0.3576	1.7963	0.3406

Table 1

Performance Measures at Individual Stations

$\sigma = 0.1$

N	λb	$N\lambda r$	m	$L\beta_m$	β_m	$\lambda\mathrm{E}[T_m]$	p_m
4	0.05	0.05	2	0.1272	0.4039	1.4756	0.5281
			3	0.2768	0.4039	1.4756	0.5094
			4	0.2813	0.2813	2.5552	0.3298
7	0.05	0.05	2	0.0353	0.2690	2.7176	0.3230
			3	0.0853	0.2696	2.7090	0.3076
			4	0.1349	0.2698	2.7061	0.2935
			5	0.1843	0.2696	2.7090	0.2804
			6	0.2337	0.2690	2.7176	0.2683
			7	0.1628	0.1628	5.1419	0.1473
12	0.05	0.05	2	0.0103	0.1634	5.1212	0.1882
			3	0.0270	0.1640	5.0985	0.1760
			4	0.0433	0.1644	5.0817	0.1649
			5	0.0591	0.1647	5.0706	0.1547
			6	0.0747	0.1649	5.0650	0.1454
			7	0.0902	0.1649	5.0650	0.1369
			8	0.1056	0.1647	5.0706	0.1292
			9	0.1212	0.1644	5.0817	0.1222
			10	0.1370	0.1640	5.0985	0.1158
			11	0.1531	0.1634	5.1212	0.1099
			12	0.0906	0.0906	10.0342	0.0559
18	0.05	0.05	2	0.0046	0.1094	8.1420	0.1368
			3	0.0124	0.1099	8.1033	0.1264
			4	0.0197	0.1102	8.0709	0.1169
			5	0.0267	0.1106	8.0444	0.1082
			6	0.0334	0.1108	8.0235	0.1003
			7	0.0400	0.1110	8.0080	0.0931
			8	0.0463	0.1111	7.9978	0.0866
			9	0.0525	0.1112	7.9927	0.0807
			10	0.0587	0.1112	7.9927	0.0753
			11	0.0648	0.1111	7.9978	0.0704
			12	0.0711	0.1110	8.0080	0.0660
			13	0.0774	0.1108	8.0235	0.0620
			14	0.0838	0.1106	8.0444	0.0583
			15	0.0905	0.1102	8.0709	0.0550
			16	0.0975	0.1099	8.1033	0.0520
			17	0.1048	0.1094	8.1420	0.0493
			18	0.0588	0.0588	15.9995	0.0245

Table 2

System Performance Measures

N	λb	$N\lambda r$	β	$\lambda \mathrm{E}[T]$
4	0.01	0.01	0.7917	0.2631
7	0.01	0.01	0.7437	0.3446
12	0.01	0.01	0.6525	0.5325
18	0.01	0.01	0.4986	1.0054
4	0.05	0.05	0.3426	1.9187
7	0.05	0.05	0.2390	3.1849
12	0.05	0.05	0.1520	5.5789
18	0.05	0.05	0.1048	8.5435

References

[1] B.D. Bunday, W.K. El-Badri and S.D. Supanekar, "The efficiency of bi-directionally patrolled machines when repairs are not always successful", European Journal of Operational Research, 19, 324-330, 1985.

[2] B.D. Bunday and E. Khorram, "The efficiency of bi-directionally patrolled machines", SOR Report, 86-8, University of Bradford, 1986.

[3] B.D. Bunday and C. Mack, "The efficiency of bi-directionally traversed machines", Journal of the Royal Statistical Society, C22, 74-81, 1973.

[4] E.G. Coffman and E.N. Gilbert, "Polling and greedy servers", Queueing Systems, 2, 115-145, 1987.

[5] E.G. Coffman and M. Hofri, "On the expected performance of scanning disks", SIAM Journal of Computation 11, 60-70, 1982.

[6] W.B. Kim and E. Koenigsberg, "The efficiency of two groups of N machines served by a single robot", Journal of the Operational Research Society, 38, 323-338, 1987.

[7] G.B. Swartz, "Analysis of a SCAN policy in a gated loop system". In Applied Probability - Computer Science: The Interface, Vol. 2, R.L. Disney and T.J. Otts, Eds. Birkhauser, Boston, 241-252, 1982.

[8] H. Takagi, "Analysis of Polling Systems", The MIT Press, Cambridge, Massachusetts, 1986.

[9] H. Takagi, "Exact analysis of round-robin scheduling of services", IBM Journal of Research and Development, 31, 484-488, 1987.

[10] H. Takagi, "Analysis of single-buffer polling systems", TRL Research Report, TR87-00 48, IBM Research Laboratory, Tokyo, 1988.

[11] H. Takagi, "A bibliography on the analysis and applications of polling models", IBM Research Laboratory, 5-19 Sanban-cho, Chiyoda-ku, Tokyo 102, Japan. Phone 81-3-265-4351; e-mail: TAKAGI at JPNTSCVM(BITNET).

[12] H. Takagi and M. Murata, "Queueing analysis of scan-type TDM and polling systems". In Computer Networking and Performance Evaluation. T. Hasegawa, H. Takagi and Y. Takahashi, Eds. Elsevier North Holland, Amsterdam, 199-211, 1986.

(Received February 3, 1990)

Acta Cybernetica, Tom. 10, Fasc. 1-2, Szeged, 1991

Heuristics for the 0-1 Min-Knapsack Problem

J. Csirik * J.B.G. Frenk M. Labbé † S. Zhang

Econometrisch Instituut, Erasmus Universiteit Rotterdam, P.O. Box 1738, 3000DR Rotterdam, the Netherlands

Abstract

The 0-1 min-knapsack problem consists in finding a subset of items such that the sum of their sizes is larger than or equal to a given constant and the sum of their costs is minimized. We first study a greedy-type heuristic having a worst-case bound of 2. This heuristic is then refined to obtain a new one with a worst-case bound of 3/2.

1 Introduction

The classical 0-1 knapsack problem (*max-knapsack*) has been extensively studied in the literature. Some greedy-type heuristics have been analysed and ε-approximation schemes are known for this problem ([2,5,6,7]). On the contrary, the *min-knapsack* problem has found until now only few interest in the English literature. Most of the results and algorithms are translated from Russian ([1,3,4]), and are given without proof.

The min-knapsack problem is formulated as follows:
given n pairs of positive integers (c_j, a_j) and a positive integer M, find $x_1, x_2, \ldots, x_n$ so as to

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} & \sum_{j=1}^{n} a_j x_j \geq M \\ & x_j \in \{0,1\}, j = 1, \ldots, n. \end{array}$$

The problem is clearly NP-hard, and so finding a good heuristic solution is of interest. Obviously, the problem is feasible if and only if $\sum_{j=1}^{n} a_j \geq M$. Next we assume that this condition is satisfied for the considered problems.

In this paper, we analyse a greedy heuristic proposed by Gens and Levner [4]. A similar heuristic also exists for the max-knapsack problem. However, for the min-knapsack, we need a slight modification of the main idea. Then, the different behaviour of this heuristic for the max and min problems is shown, when the item sizes a_j are bounded by $M/k, k \geq 2$. The heuristic we consider has a worst-case bound of 2. We then provide a refinement with worst-case bound of 3/2, with a possible ε-approximation scheme extension. We finally propose some practical improvement.

*On leave from Department of Computer Science, University of Szeged, Árpád tér 2, 6720 Szeged, Hungary

†Fellow of the European Institute for Advanced Studies in Management, Brussels

2 The Heuristic

We will use throughout a_j to denote an *item* as well as its *size*, while c_j represents its *cost*. Furthermore, c_j/a_j is defined as the relative cost of item a_j.

Algorithm *GR*

Step 1. Sort the items in nondecreasing order of their relative costs. From now on, we assume that

$$c_1/a_1 \leq c_2/a_2 \leq \cdots \leq c_n/a_n.$$

Step 2. (a) Let k_1 be the index for which

$$\sum_{i=1}^{k_1} a_i < M \leq \sum_{i=1}^{k_1+1} a_i.$$

Then the sublist $(a_1, a_2, \ldots, a_{k_1+1})$ is a candidate for the solution given by the heuristic GR. Let

$$S_1 = (a_1, a_2, \ldots, a_{k_1}),$$

then this candidate can be written as

$$S_1 \cup \{a_{k_1+1}\}.$$

(b) Let $k_1 + 2, k_1 + 3, \ldots, k_2 - 1$ be a (possibly empty) series of indices so that for all of the corresponding items (i.e. $j \in \{k_1 + 2, \ldots, k_2 - 1\}$) the following holds:

$$\sum_{i=1}^{k_1} a_i + a_j \geq M.$$

Let

$$B_1 = (a_{k_1+1}, \ldots, a_{k_2-1}),$$

then all $S_1 \cup \{a_j\}, j \in \{k_1 + 2, \ldots, k_2 - 1\}$, are also candidate solutions.

(c) Now, let k_2 be the first next index for which

$$\sum_{i=1}^{k_1} a_i + a_{k_2} < M,$$

and let $k_3 \geq k_2$ be the index for which

$$\sum_{i=1}^{k_1} a_i + \sum_{i=k_2}^{k_3} a_i < M \leq \sum_{i=1}^{k_1} a_i + \sum_{i=k_2}^{k_3+1} a_i.$$

Set

$$S_2 = (a_{k_2}, a_{k_2+1}, \ldots, a_{k_3}).$$

Then, $S_1 \cup S_2 \cup \{a_{k_3+1}\}$ is also a candidate solution.

Now iterate from (b), with, in the first iteration, k_3 instead of k_1 and k_4 instead of k_2; in the i-th iteration, use k_{2i+1} and k_{2i+2}. Repeat this until the end of the list. The solution given by heuristic GR is the minimum cost candidate. It is easy to see that Steps 1 and 2 have a computational complexity of $O(n \log n)$ and $O(n)$ respectively.

3 Results

Let us denote the cost of an fixed optimal solution $\overline{X}$ for the list

$$L = (a_1, a_2, \ldots, a_n)$$

by $OPT(L)$, and the cost of the solution given by heuristic GR for the same list by $GR(L)$.

Lemma 1 *For all lists L,*

$$GR(L) \leq 2 \cdot OPT(L).$$

Proof. By applying heuristic GR to list L, we subdivise L into a sequence of alternating sublists as follows:

$$\overbrace{a_1, a_2, \ldots, a_{k_1}}^{S_1}, \overbrace{a_{k_1+1}, \ldots, a_{k_2-1}}^{B_1}, \overbrace{a_{k_2}, \ldots, a_{k_3}}^{S_2}, \overbrace{a_{k_3+1}, \ldots, a_{k_4-1}}^{B_2}, \ldots,$$

$$, \ldots, \overbrace{a_{k_{2\ell-1}+1}, \ldots, a_{k_{2\ell}-1}}^{B_\ell}, \overbrace{a_{k_{2\ell}}, \ldots, a_{k_{2\ell+1}}}^{S_{\ell+1}}, \ldots,$$

$$, \ldots, \overbrace{a_{k_{2m-1}+1}, \ldots, a_{k_{2m}-1}}^{B_m}, \overbrace{a_{k_{2m}}, \ldots, a_n}^{S_{m+1}}$$

where the last set is possibly empty, in which case $k_{2m} - 1 = n$.

Let us call the elements in S-lists *small* and in B-lists *big*. Then, clearly, the heuristic solution has exactly one big element and contains all small elements before this big element. Furthermore, it is the cheapest solution among all such candidates.

From the algorithm, we have:

$$\sum_{a_i \in \cup_{j=1}^{\ell} S_j} a_i + a_r \geq M, \text{for all } \ell = 1, \ldots, m-1 \text{ and all } a_r \in B_\ell, \tag{1}$$

and

$$\sum_{a_i \in \cup_{j=1}^{\ell} S_j} a_i < M, \text{for all } \ell = 1, \ldots, m+1. \tag{2}$$

Since inequality (2) holds in particular for $\ell = m+1$, we know that the optimal solution contains at least one big element. Let a_t be the big element with smallest index in the optimal solution and let B_q be the set containing a_t. From the algorithm, we know that

$$GR(L) \leq \sum_{a_i \in \cup_{j=1}^{q} S_j} c_i + c_t.$$

Now, let $J^* = \{i : 1 \leq i \leq n \& \overline{X}_i = 1\}$, $I = \{i : a_i \in \cup_{j=1}^q S_j\}$, $J = \{i : i \in I \& \overline{X}_i = 1\}$, $K = \{i : i \in I \& \overline{X}_i = 0\}$. Then $I = J \cup K$ and $J \cap K = \emptyset$. Since all items a_i with $i < t$ have a relative cost not larger than c_t/a_t we obtain

$$\sum_{\substack{a_i \in \cup_{j=1}^q S_j \\ i \in I}} c_i + c_t = \sum_{\substack{a_i \in (\cup_{j=1}^q S_j) \cap L^{OPT} \\ i \in J}} c_i + \sum_{\substack{a_i \in (\cup_{j=1}^q S_j) \setminus L^{OPT} \\ i \in K}} c_i + c_t \tag{3}$$

$$\leq \sum_{\substack{a_i \in (\cup_{j=1}^q S_j) \cap L^{OPT} \\ i \in J}} c_i + \frac{c_t}{a_t} \sum_{\substack{a_i \in (\cup_{j=1}^q S_j) \setminus L^{OPT} \\ i \in K}} a_i + c_t.$$

Applying (2) to the second term in the above inequality yields that the upperbound in (3) is bounded from above by

$$\sum_{\substack{a_i \in (\cup_{j=1}^q S_j \cap L^{OPT} \\ i \in J}})c_i + \frac{c_t}{a_t}(M - \sum_{\substack{a_i \in (\cup_{j=1}^q S_j) \cap L^{OPT} \\ i \in J}} a_i) + c_t$$

and this implies by the feasibility of $\overline{X}$ that

$$\sum_{\substack{a_i \in \cup_{j=1}^q S_j \\ i \in I}} c_i + c_t \leq \sum_{\substack{a_i \in (\cup_{j=1}^q S_j) \cap L^{OPT} \\ i \in J}} c_i + \frac{c_t}{a_t} \sum_{\substack{a_i \in L^{OPT} \setminus (\cup_{j=1}^q S_j) \\ i \in J^* - J}} a_i + c_t. \tag{4}$$

Finally, since the first big element in the optimal solution is $a_t \in B^q$ and hence all items in $J^* - J$ have a relative cost not smaller than c_t/a_t, we obtain by (4) that

$$\sum_{\substack{a_i \in \cup_{j=1}^q S_j \\ i \in J}} c_i + c_t \leq \sum_{\substack{a_i \in (\cup_{j=1}^q S_j) \cap L^{OPT} \\ i \in J}} c_i + \sum_{\substack{a_i \in L^{OPT} \setminus (\cup_{j=1}^q S_j) \\ i \in J^* - J}} c_i + c_t \tag{5}$$

$$= OPT(L) + c_t,$$

and using $\overline{X}_t = 1$ the expression in (5) is bounded above by $2 \cdot OPT(L)$.

It is easy to show that the bound given in Lemma 1 is tight. Consider the list $L = (1, M-2, M-1)$ with relative costs of $(1, 1, 1)$. Then, $OPT(L) = M, GR(L) = 2M - 2$ and that yields that $GR(L)/OPT(L)$ can be arbitrarily close to 2.

It is interesting to note that, contrary to the max-knapsack greedy heuristic, this bound remains the same if the items are small. Let k be a positive integer such that $k \geq 2$ and $k \ll M$. Assume that $a_i \leq M/k$ for all items, and let

$$L = (\underbrace{M/k, \ldots, M/k}_{(k-1) times}, M/k - 1, M/k)$$

with costs of

$$(\underbrace{1,\ldots,1}_{(k-1)times}, M/k-1, M/k).$$

Then, $OPT(L) = M/k+k-1$ and $GR(L) = 2M/k+k-2$. Hence, $GR(L)/OPT(L)$ can be arbitrarily close to 2 if M is large enough.

Using GR, we can derive a better heuristic as follows. For all big items $a_i \in B_L = \cup_{j=1}^{m} B_j$, we define a new knapsack problem. Let $L_i = L \setminus \{a_i\}$ and let the capacity of the knapsack $M_i = M - a_i$. The **improved heuristic** IGR is: for all $a_i \in B_L$, apply GR to the problem defined by L_i and knapsack capacity M_i. Let

$$IGR_i = GR(L_i) + c_i.$$

Then, the cost of the solution obtained with IGR is

$$IGR(L) = \min\{\min_{a_i \in B_L} IGR_i, GR(L)\}. \tag{6}$$

Since $|B_L| = O(n)$ and since, once the items are ranked by order of nonincreasing relative costs, the application of GR for each big item can be performed in linear time, the time complexity of IGR is $O(n^2)$.

Lemma 2 *For all lists L,*

$$IGR(L) \leq 3/2 \cdot OPT(L).$$

Proof. Let a_t be the smallest-index big item in a fixed optimal solution $\overline{X}$. We distinguish two cases.

(a) $c_t < 1/2 \cdot OPT(L)$.
In this case it follows directly from the proof of Lemma 1, that

$$GR(L) \leq 3/2 \cdot OPT(L),$$

and the result follows from (6).

(b) $c_t \geq 1/2 \cdot OPT(L)$.
In this case we obtain by (6) that

$$IGR(L) \leq GR(L_t) + c_t$$

and hence by the worst-case result for $GR(L)$ mentioned in Lemma 1

$$IGR(L) \leq 2 \cdot OPT(L_t) + c_t.$$

Observing now that $\overline{X}_t = 1$ finally yields

$$IGR(L) \leq 2 \cdot OPT(L_t) + c_t \leq 2 \cdot (OPT(L) - c_t) + c_t \leq 3/2 \cdot OPT(L).$$

We could get heuristics with better and better worst-case bounds by applying successively the improved method to pairs, triplets,... of big elements. This would lead to a heuristic similar to the one given by Sahni [7] for the max- knapsack problem. The result of this series of improvements is a polynomial approximation scheme, which is not fully polynomial.

From a practical point of view, we can improve the behaviour of GR, without changing its worst-case bound. This improved heuristic will be called GR^+ and consists in the following. Let $a'_1, a'_2, \ldots, a'_k$ be a candidate solution for GR (i.e. $a'_1, a'_2, \ldots, a'_{k-1}$ are small and a'_k is a big item). We delete a'_{k-1} if

$$\sum_{i=1}^{k-2} a'_i + a'_k \geq M.$$

If we could delete a'_{k-1}, then we try to delete a'_{k-2}. This is possible if

$$\sum_{i=1}^{k-3} a'_i + a'_k \geq M.$$

Hence, we delete items until

$$\sum_{i=1}^{\ell-1} a'_i + a'_k < M.$$

and the candidate solution for GR^+ is:

$$a'_1, a'_2, \ldots, a'_\ell, a'_k.$$

The solution given by GR^+ is the minimum cost candidate.

Clearly, the candidates for GR^+ are not more expensive than the candidates for GR, so that

$$GR^+(L) \leq GR(L) \leq 2 \cdot OPT(L).$$

References

[1] Babat L.G., Linear functionals on the n-dimensional unit cube, *Soviet. Math. Dokl.* *16*(1975), 398-400.

[2] Friese A.M. and M.R.B. Clarke, Approximation algorithms fot the m- dimensiomal 0-1 knapsack problem: worst-case and probabilistic analyses, *Eur. J. Oper. Res.* *15*(1984), 100-109.

[3] Gens G.B. and Y.V. Levner, Approximate algorithms for certain universal problems in scheduling theory, *Engineering and Cybernetics* *3*(1978),31-36.

[4] Gens G.B. and Y.V. Levner, Computational complexity of approximation algorithms for combinatorial problems,*Lecture Notes in Computer Sciences, Vol. 74*, Mathematical Foundations of Computer Science,292- 300.

[5] Ibarra O.H. and C.E. Kim, Fast approximation algorithms for the knapsack and sum of subset problems, *JACM* *22*(1975), 463-468.

[6] Lawler E.L., Fast approximation algorithms for knapsack problems, *Math. of O.R.* *4*(1979), 339-356.

[7] Sahni S., Approximate algorithms for the 0-1 knapsack problem, *JACM* *22*(1975), 115-124.

(Received May 20, 1991)

Acta Cybernetica, Tom. 10, Fasc. 1-2, Szeged, 1991

A 1.6 lower-bound for the two-dimensional on-line rectangle bin-packing

G. Galambos*

Abstract

Examining on-line algorithms for the two dimensional rectangle bin packing problem, Coppersmith asked in [2] whether one can give a better lower bound for this type of algorithms than the Liang's bound which is 1.5364... . In this paper we present a bound of 1.6.

Keywords: two-dimensional bin-packing, worst-case analysis, on-line algorithms, heuristic.

1 Introduction

Let us first consider the one-dimensional bin-packing problem: There is given a list $L(n) = \{a_1, \dots a_n\}$, and let us suppose that a size l_i belongs to each $a_i \in L(n)$, $(0 < l_i \leq 1)$. The problem is to pack the elements of $L(n)$ into unit-capacity bins, while attempting to minimize the number of bins needed for packing. The problem is NP-hard (cf.[3]) and therefore various heuristic algorithms have been studied for solving this problem. Let us consider the following class of approximation algorithms which produce a near-optimal solution of the problem: An algorithm belonging to this class packs the elements one by one in the order given by list $L(n)$, and after having placed the element into the bin, it will be never moved again. The algorithms belonging to this class are called on-line algorithms.

One possibility to measure the performance of an algorithm A is to give its asymptotic worst case performance ratio R_A : Let L be a list and denote by L^* the minimal number of bins needed to pack the list L. Moreover, let $A(L)$ represent the number of bins that are used by the algorithm A to pack the elements of L. If $R_A(k)$ denotes the supremum of the ratios $A(L)/L^*$ for all lists L with $L^* = k$, then

$$R_A = \limsup_{k \to \infty} R_A(k).$$

The best known lower bound for R_A for the class of on-line algorithms has heen given by Liang (cf.[4]). He proved that there is no on-line algorithm A for which $R_A < 1.5364...$ .To verify this result Liang considered a $k \in \mathbb{N}_+$ and defined the sequence $m_0 = 1, m_{j+1} = m_j(m_j + 1), (1 \leq j \leq k)$. Finally, he considered the lists $L_k, \dots, L_0$ where L_j $(0 \leq j \leq k)$, represents a block of n elements of size $l_j = \frac{1}{m_j+1} + \epsilon$ with $\epsilon > 0$ suitable small chosen. He proved for the concatenated

*Department of Applied Computer Science József Attila University, Szeged Hungary.

lists $L_k, L_kL_{k-1}, \ldots, L_k \ldots L_0$ that one of the ratios $A(L_k \ldots L_j)/(L_k \ldots L_j)^*, 0 \leq j \leq k$, is at least 1.5364... for every n.

Now let us consider the following two-dimensional generalization of the one-dimensional problem (cf.[1]): We are given a list of $L(n) = \{a_1, \ldots a_n\}$ with an ordered pair of sizes $(w(a_i), h(a_i))$ where $w(a_i)$, resp. $h(a_i)$ is the width, resp. height of a_i and we are given rectangular bins with sizes W and H. (Without loss of generality we can suppose that $W = H = 1$ and $w(a_i) \leq 1, h(a_i) \leq 1$.) We have to pack the rectangles into the minimal number of bins such that

- the sides of the elements are parallel to the corresponding sides of the bins (no rotation allowed);

- no two rectangles in a bin overlap.

The definition of an on-line algorithm for the two-dimensional case is the same as for the one-dimensional case. It is very easy to see for the lists L_j satisfying $h(a_j) = \frac{1}{m_j+1} + \epsilon$ and $w(a_j) = 1$ that we get a trivial lower bound for the asymptotic worst case ratio of an arbitrary two-dimensional on-line algorithm. This means that the following theorem is true:

Theorem 1 *There is no on-line two-dimensional bin-packing algorithm A for which $R_A < 1.5364....$*

In [2] Coppersmith mentioned that no better lower bound is known. In this paper we restate the trivial argument and obtain a slightly improved, but non-trivial, lower bound of 1.6.

2 Computation of lower bound

In order to prove the lower bound, we introduce the following lists. Let k be an arbitrary integer, we choose $n = 4k$, and consider the lists L_1, L_2, L_3, L_4 with

- L_1 contains n pieces of A-elements with sizes $(\frac{1}{2} - \epsilon, \frac{1}{2} - 2\epsilon)$;
- L_2 contains n pieces of B-elements with sizes $(\frac{1}{2} + \epsilon, \frac{1}{2} - \epsilon)$;
- L_3 contains n pieces of C-elements with sizes $(\frac{1}{2} - 2\epsilon, \frac{1}{2} + 2\epsilon)$;
- L_4 contains n pieces of D-elements with sizes $(\frac{1}{2} + 2\epsilon, \frac{1}{2} + \epsilon)$;

Lemma 1

$$(L_1 \ldots L_j)^* \leq j\frac{n}{4}. \qquad j = 1, 2, 3, 4.$$

Proof. We leave it to the reader to verify the cases $j = 1, 2, 4$, and we prove only the case $j = 3$:
We give a feasible packing which consists of the following bins:

- $\frac{n}{4}$ times bins with two A-elements and two C-elements;

- $\frac{n}{2}$ times bins with $1A$-elements, $2B$-elements an $1C$-elements. (see Figure 1.) □

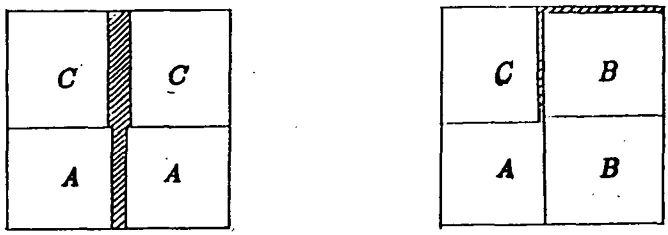

Figure 1: A possible packing of (L_1, L_2, L_3).

Let us now pack the elements of the concatenated list $(L_1 L_2 L_3 L_4)$. We say that a bin has type $\mathbf{t} = (t_1, t_2, t_3, t_4)$ if it contains t_1 pieces of A-elements, t_2 pieces of B-elements etc., and we denote the set of bins after having packed the concatenated list by T. Moreover, if a bin is represented by its content (t_1, t_2, t_3, t_4) we define the following subsets:

$$T_k = \{(t_1, t_2, t_3, t_4) \in T | i_j = 0, \text{ if } j < k \text{ and } i_k > 0\}. \quad 1 \leq k \leq 4.$$

It is clear that $T = \bigcup_{i=1}^{4} T_i$ and $T_i \cap T_j = \emptyset$ if $i \neq j$.

We denote by $a(\mathbf{t})$ the number of bins which contain t_j elements from the list L_j, $(1 \leq j \leq 4)$.

Now we are ready to state our lower bound theorem:

Theorem 2 *There is no on-line two-dimensional bin-packing algorithm A for which $R_A < 1.6$.*

Proof.: We examine how many bins have been used after having packed the list L_j, $(1 \leq j \leq 4)$:

$$A(L_1 \dots L_j) = \sum_{i=1}^{j} \sum_{\mathbf{t} \in T_i} a(\mathbf{t}) \tag{1}$$

and the number of the packed elements for each j, $\quad 1 \leq j \leq 4$:

$$n = \sum_{\mathbf{t} \in T} t_j a(\mathbf{t}), \qquad 1 \leq j \leq 4. \tag{2}$$

Adding all the equations (1) and substracting (2) it follows:

$$A(L_1) + A(L_1 L_2) + A(L_1 L_2 L_3) + A(L_1 L_2 L_3 L_4) - 4n =$$

$$4 \sum_{\mathbf{t} \in T_1} a(\mathbf{t}) + 3 \sum_{\mathbf{t} \in T_2} a(\mathbf{t}) + 2 \sum_{\mathbf{t} \in T_3} a(\mathbf{t}) + \sum_{\mathbf{t} \in T_4} a(\mathbf{t}) - \sum_{\mathbf{t} \in T} a(\mathbf{t}) \sum_{j=1}^{4} t_j. \tag{3}$$

Lemma 2 *The right hand side of (3) is non negative.*

Proof. First we consider a bin which is in T_1. We have to prove that for each such type of bin $\sum_{i=1}^{4} t_i \leq 4$. In other words, we have to prove that in each bin in which there is at least one element of L_1 the maximum number of the elements is 4. And this is trivial.

Similarly we have to prove obvious statements in the other cases as well.

□

We introduce the following notations

$$r_j = \frac{A(L_1 \dots L_j)}{(L_1 \dots L_j)^*}, \qquad 1 \leq j \leq n. \tag{4}$$

and

$$r = \max_{1 \leq j \leq 4} r_j. \tag{5}$$

Now using the Lemmas 1-2 and replacing (4) into the left hand side of (3) we get

$$\sum_{j=1}^{4} j r_j \geq 16.$$

Now using (5) we get the statement of our theorem.

3 Conclusions

Since the best known on-line algorithm has been analysed in [5], and its asymptotic worst case ratio is about 2.86, the gap between the given lower-bound and this value is large. On the one hand we are sure that this very simple construction studied in our paper has a refinement, and we suspect a lower bound near to 2. On the other hand one can show that the examined algorithms do not used out deeply that our problem is "two-dimensional" and most of them are different generalizations of the known - and analysed - one-dimensional algorithms. So we suspect that with a new method the researchers will be able to give better algorithms than the Generalyzed Harmonic Fit which was presented in [5].

References

[1] Chung, R.F.K., Garey, M.R., Johnson, D.S.: On packing two-dimensional bins. SIAM Alg. Discr. Meth. 3, 66-76 (1982).

[2] Coppersmith, P.R., Raghavan, P.: Multidimensional on-line bin-packing: algorithms and worst-case analysis. Op. Res. Letters 8, 17-20 (1989).

[3] Garey, M.R., Johnson, D.S.: Computer and Intractibility: A Guide to the theory of NP-Completeness. San- Francisco: W.H. Freeman 1979.

[4] Liang, F. M.: Lower bound for on-line bin-packing. Inf. Proc. Letters 10, 76-79 (1980).

[5] Li, K., Cheng, K-H.: A Generalized Harmonic Algorithm for On-Line Multi-Dimensional Bin Packing. Research Report, Dep. of Computer Science, Houston.

(Received November 1, 1990)

Acta Cybernetica, Tom. 10, Fasc. 1-2, Szeged, 1991

On strict codes

Nguyen Huong Lam and Do Long Van *

Abstract

This paper continues an earlier paper of the authors. The maximality, the decomposability etc. for infinitary strict codes are considered. Every infinitary strict code is shown to be included in an infinitary strict-maximal one. The so-called Theorem of Defect for infinitary strict codes is proved. Some conditions for an infinitary strict code to be written by an indecomposable ones are stated.

1 Preliminaries

The concept of strict codes has been first mentioned in [3]. The classical definition of a code says that the (finite) identity relations are the only (finite) relations satisfied by the code. For a strict code the demand is stronger: any relation, finite or infinite, which is satisfied by the code is an identity one. So, strict codes form a subclass of codes. In [7] a particular case of strict codes, namely, that of finitary strict codes was considered. In our paper [6] we studied infinitary strict codes; we proposed some procedures to verify whether a given language is a strict code. Also, we characterized strict codes by ∞-submonoids generated by them. In the present article, which is a sequel to [6], we mostly adapt some well-known notions and properties of codes for strict one such as maximality, decomposability, Theorem of Defect, etc.

In what follows we mostly use standard terminology and notation (see, e.g. [5], [1]). Let A be an alphabet, finite or countable, A^* the free monoid generated by A whose elements are called finite words. We denote A^N the set of infinite words over A and $A^\infty = A^* \cup A^N$ whose elements we call simply words. We make A^∞ a monoid equipping it with the product defined as:

For any words α, β of A^∞,

$$\alpha\beta = \begin{cases} \alpha & \text{if } \alpha \in A^N, \beta \in A^\infty \\ \alpha\beta & \text{if } \alpha \in A^*, \beta \in A^\infty \end{cases}$$

where $\alpha\beta$ means the catenation of α and β (see [6]). Clearly, the empty word, denoted by ϵ, is the unit of A^∞.

We call a subset X of A^∞ (respectively, of A^*) an infinitary language (respectively, finitary language). For a finite subset X, Card$X$ denotes its cardinality; also, to simplify the notation we often identify a singleton set with its element. For a word $x \in A^*, |x|$ denotes the length of x and we say by convention that $|\epsilon| = 0$ and $|x| = \omega$ if $x \in A^N$.

*Institute of Mathematics, P.O.Box 631, Bo Ho, Hanoi, Vietnam

For any infinitary language X we denote:

$X_{\text{fin}} = X \cap A^*, X_{\text{inf}} = X \cap A^N$

$XY = \{\alpha\beta : \alpha \in X, \beta \in Y\}$, $Y \subseteq A^\infty$

(the product extended to languages).

$X^2 = XX$

$X^{n+1} = XX^n$, $n = 1, 2, \ldots$

$X^* = \bigcup_{n \geq 0} X^n$

(the smallest submonoid of A^∞ containing X).

$X^+ = X^* - \epsilon$

$X^\omega = \{x_1 x_2 \ldots : x_i \in X_{\text{fin}}, i = 1, 2, \ldots\}$

(the set of all infinite products of elements in X_{fin}).

$X^\infty = X^* \cup X^\omega$

$X^{+\infty} = X^\infty - \epsilon.$

For every $n \geq 1$ we introduce the set $X_{(n)}$ of n-tuples defined as:

$X_{(n)} = \{(x_1, x_2, \ldots, x_n) : x_1, \ldots, x_{n-1} \in X_{\text{fin}}, x_n \in X\}$

and the set $X_{(\omega)}$ of ω-tuples defined as:

$X_{(\omega)} = \{(x_1, x_2, \ldots) : x_i \in X_{\text{fin}}, i = 1, 2, \ldots\}$

and we put

$X_{(*)} = \bigcup_{n \geq 1} X_{(n)}$

$X_{(\infty)} = X_{(*)} \cup X_{(\omega)}$

We say that a word x of A^∞ admits a *$*$-factorization* (resp. an *ω-factorization*) $(x_1, x_2, \ldots)$ over X provided $x = x_1 x_2 \ldots$ with $(x_1, x_2, \ldots) \in X_{(*)}$ (resp. $(x_1, x_2, \ldots) \in X_{(\omega)}$); we say that X admits an *∞-factorization* over X if it admits either a $*$-factorization or an ω-factorization over X. A given subset X is said to be an *infinitary code* (resp. an *infinitary strict code*) if every word of A^∞ admits at most one $*$-factorization (resp. one ∞- factorization) [6].

Finally, for any two subsets X, Y of A^∞, we define:

$XY^{-1} = \{\alpha \in A^\infty : \exists \beta \in Y : (\alpha\beta \in X) \wedge ((|\alpha| = \omega) \Longrightarrow \beta = \epsilon)\}$.

$Y^{-1}X = \{\alpha \in A^\infty : \exists \beta \in Y : (\beta\alpha \in X) \wedge ((|\beta| = \omega) \Longrightarrow \alpha = \epsilon)\}$.

2 Maximality

In this section we consider maximality properties of strict codes and ∞-submonoids generated by them. First, we show that each strict code is included in a strict-maximal one. A strict code X is called *strict-maximal* if it is not contained properly in any other strict code. X is called *relatively strict-maximal* if for every finite word $w \in A^*, X \cup w$ is no more a strict code.

Theorem 2.1 *Every strict code is contained in a strict-maximal one over A.*

Proof: First, we prove that every strict code is included in a relatively strict-maximal one. To do this we enumerate all finite nonempty words in some order

$$A^+ = \{w_1, w_2, \ldots\}$$

and define an increasing sequence of strict codes $X_0 \subseteq X_1 \subseteq X_2 \subseteq \ldots$ as follows:

Put $X_0 = X$ and suppose for some $n \geq 0, X_n$ has been defined. Let $i(n)$ be the smallest integer such that $X_n \cup w_{i(n)}$ is a strict code (if no such $i(n)$ exists, we put $X_{n+1} = X_n$). Put $X_{n+1} = X_n \cup u_n$, where u_n is any word in $\{X_n \cup w_{i(n)}\}^* w^\omega_{i(n)}$. Since $X_n \cup w_{i(n)}$ is a strict code, so is X_{n+1}. Thus X_{n+1} is defined and by induction our sequence is built.

Consider the set $Y = \bigcup_{n\geq 1} X_n$. Since $Y = X \cup \{u_n : n \geq 0\}$ and all u_n's are infinite words it follows that every ∞-factorization over Y is also one over X_n for some n. This means that Y is a strict code because each X_n is a strict code. Also, it is easy to see that Y is relatively strict-maximal by construction of the sequence X_n. Thus every strict code is contained in a relatively strict-maximal one.

Next, we prove that the class of relatively strict-maximal codes is inductive, i.e. every chain (by inclusion) has an upper bound. Indeed, let

$$X_\alpha \subseteq X_\beta \subseteq X_\gamma \subseteq \dots$$

be a chain in this class, indexed by a set I. Since each member of this chain is relatively strict-maximal, we have

$$X_{\alpha\mathrm{fin}} = X_{\beta\mathrm{fin}} = X_{\alpha\mathrm{fin}} = \dots$$

Putting

$$Z = \bigcup_{\gamma\in I} X_\gamma$$

we have $Z_{\mathrm{fin}} = X_{\gamma\mathrm{fin}}$ for every $\gamma \in I$ that means Z is relatively strict-maximal and thus Z is an upper bound of the chain. So, by Zorn's Lemma, every relatively strict-maximal code is included in a maximal one, but it is easy to see that every maximal element of this class is a strict-maximal code. Theorem is proved.

It has been known that there exists an algorithm to decide whether a given finitary language is a maximal code (in the class of finitary codes) [1]. Below we state a similar result for finitary strict codes. Recall that a finitary language X is said to be *complete* if for any w of A^* : $A^*wA^* \cap X^* \neq \emptyset$. A word w of A^* is called *overlapping* if $w = ux = yu$ for some $u, x, y \in A^+$. It is easy to see that for every w of A^* (Card $A \geq 2$) there exist x and y of A^* such that xwy is not overlapping. Consider now in the class of finitary strict codes a maximal one by inclusion. We call such a code *finitary-strict-maximal,* or, following just defined terminology, X is finitary relatively strict-maximal code. We have then:

Proposition 2.2 *Every finitary-strict-maximal code X is complete.*

Proof: Let $w \in A^* - X$. As noted $xwy = w'$ is not overlapping for some x, y in A^*. If $w' \in X^*$, nothing is to be proved. Suppose $w' \notin X^*$, then $X \cup w'$ is not a strict code, it has to exist an infinite equality

$$x_1x_2\dots = y_1y_2\dots$$

over $X \cup w'$ with $x_1 \neq y_1$. Let $x_k = w', k \geq 1$ and m and n be respectively the largest and the smallest integers such that $x_k = w'$ is a subword of $y_my_{m+1}\dots y_n$. Since w' is not overlapping, there is no w' among $y_m, \dots, y_n$ meaning that they are all in X. Consequently, w' is a subword of $y_my_{m+1}\dots y_n \in X^*$ and so is w. This completes the proof.

As a consequence of Proposition 2.2., we have

Theorem 2.3 *There exists an algorithm to decide whether a given finitary finite subset X is a finitary-strict-maximal code.*

Proof: First, by Theorem 2.6. of [6], we can verify whether X is a strict code. Second, if X is finitary-strict-maximal code it must be complete. But it has been known that a finite (strict) code is complete if and only if it is maximal as a code (and therefore finitary-strict-maximal as a strict code). Thus it suffices to test the maximality of X as a code and this could be done, for example, by means of a Bernoulli distribution. Because X is finite the test is always effective. Theorem is proved.

We now return to the general framework of infinitary words and languages. We state some properties of strict-maximal codes analogous to the case of ordinary ones, but let us first define some notions.

A language $X \subseteq A^\infty$ is said to be *dense* if for every α of $A^\infty, A^\infty \alpha A^\infty \cap X \neq \emptyset$, or, which amounts to the same, for every w of A^N : $A^* w \cap X_{\text{inf}} \neq \emptyset; X$ is said to be *complete* (resp. *weakly complete*) if X^* (resp. X^∞) is dense.

Theorem 2.4 *Every strict-maximal code is weakly complete.*

Proof.: If the alphabet A is a singleton $A = \{a\}$ then a language X is a strict code if and only if $X = \{a^n\}$ for some positive interger n or $X = \{a^\omega\}$. So every strict code is strict-maximal and weakly complete in this case.

Suppose now Card$A \geq 2$ and X is a strict-maximal code. We prove that $A^*\alpha \cap X_{\text{inf}} \neq \emptyset$ for every α in A^N. If $\alpha \in X_{\text{inf}}$, we are done, otherwise $X \cup \alpha$ is not a strict code, so that we have an equality:

$$x_1 x_2 \ldots = y_1 y_2 \ldots \tag{1}$$

with two possibilities:

(i) $(x_1, x_2, \ldots) \in (X \cup \alpha)_{(n)}$ and $(y_1, y_2, \ldots) \in (X \cup \alpha)_{(m)}$ for $m, n \geq 1$.

(ii) $(x_1, x_2, \ldots) \in X_{(\omega)}$ and $(y_1, y_2, \ldots) \in (X \cup \alpha)_{(m)}$.

If (ii) holds, we are done. Now suppose that we have (i). If $x_n \neq \alpha$, we are through again, otherwise, from (1) and from the fact that α must occur among y_i's, we have $y_m = \alpha$. Hence $\alpha = p^\omega$ for some primitive word $p \in A^*$. We choose a letter b different from the last letter of p. Consider the word $bp^\omega = b\alpha$, which we can suppose not to belong to X (in the contrary, we are done), therefore the set $X \cup bp^\omega$ is not a strict code. But now with bp^ω playing the role of α, we have the equality (1) for $X \cup bp^\omega$. The case (i) with $x_n = bp^\omega = b\alpha = y_m$ is already impossible, otherwise $bp^\omega = q^\omega$ for some another primitive word q. Certainly $q = bq', q' \in A*$, hence $p^\omega = (q'b)^\omega$. Since $q'b$ is also primitive, we get $p = q'b$ which contradicts the fact that the last letter of p is not b. So we have now either (i) with $x_n \neq b\alpha, y_m = b\alpha$ (or $x_n = b\alpha, y_m \neq b\alpha$) or (ii). Consequently, $A^* b\alpha \cup X \neq \emptyset$ and the theorem follows.

It is well-known that every recognisable complete finitary code is a maximal one, but we cannot state such an analog for strict codes as shown in the next example.

Example 2.5 *Consider the language $X = \{a^2, bA^\omega\}$ over the binary alphabet $A = \{a, b\}$. It is easy to see that X is a recognizable weakly complete strict code (even more so, complete one), but not a strict-maximal code, since $X \cup abA^\omega$ is still a strict code.*

Given two languages X, Y of A^∞, X is said to be *written* by Y, in notation $X \prec Y$, if $X \subseteq Y^\infty$ and no proper subset Z of Y has this properties i.e. $\forall Z \subset Y, X \not\subseteq Z^\infty$. It is easy to verify that in the class of strict codes $\prec$ is a partial order. A strict code X is said to be *indecomposable* over A if for any strict code $Y, X \prec Y$ implies $Y = X$ or $Y \subseteq A$.

Example 2.6 *(i) Consider the strict code $X = \{a^2, b, (ba)^\omega\}$ over the alphabet $A = \{a, b\}$; X is not indecomposable because $X \prec \{a^2, b, (ab)^\omega\}$.*

(ii) Consider now the strict code $X = \{b^3a, b^2, a, abab^\omega\}$. We show that X is indecomposable. Indeed, if X is written by a strict code $Y \not\subseteq A$ then Y_{fin} equals either $\{b^3a, b^2, a\}$ or $\{ba, b^2, a\}$, and $abab^\omega = y_1y_2$ for some $y_1 \in Y^*_{\text{fin}}, y_2 \in Y_{\text{fin}}$ (the case $abab^\omega \in Y^\omega_{\text{fin}}$ is impossible). If $Y_{\text{fin}} = \{b^3a, b^2, a\}$ then y_1 must be ϵ, otherwise $y_2 = bab^\omega$, which is impossible because we would have then $b^3a(b^2)^\omega = b^2y_2$. Hence $abab^\omega = y_1y_2 = y_2 \in Y$ meaning that $Y = X$.

If now $Y_{\text{fin}} = \{ba, b^2, a\}$, then $y_1 = \epsilon, y_1 = a$ or $y_1 = aba$, correspondingly $y_2 = abab^\omega, y_2 = bab^\omega$, or $y_2 = b^\omega$. In all cases, it is easy to see that Y is not a code, which is a contradiction.

We recall some notations introduced in [6]. A subset M of A^∞ is called ∞-*submonoid* if $M^\infty = M$ and it is called freeable if $M^{-1}M \cap MM^{-1} = M$. Every subset X of an ∞-submonoid M such that $X^\infty = M$ is c alled a *generator set* of M. It was proved in [6] that every ∞-submonoid possesses a smallest generator set in the sense that it is contained in every generator set of M and freeable ∞-submonoids are always generated by strict codes which are the smallest generator sets of them. An ∞-submonoid of A^∞ is said to be *freeable-maximal* provided it is not contained in any freeable ∞-submonoid other than itself and A^∞.

We state now a result relating indecomposability off a strict code and freeable-maximality of the ∞-submonoid generated by it.

Theorem 2.7 *An ∞-submonoid M is freeable-maximal if and only if it is generated by an indecomposable strict-maximal code.*

Proof.: Let M be freeable-maximal and X its smallest generator set which is a strict code. If X is not strict-maximal, then there exists a word x of $A^\infty - X$ such that $X \cup x$ is a strict code. Hence $M = X^\infty \subset (X \cup x)^\infty$ which implies $(X \cup x)^\infty = A^\infty$. Since $X \cup x$ is a strict code, we have $X \cup x = A$, therefore x belongs to A. On the other hand $X \cup x^\infty$ is also a strict code and we have

$$M \subset (X \cup x^\omega)^\infty \subset A^\infty$$

i.e. M is not freeable-maximal: a contradiction. That X is not indecomposable also leads to a contradiction. Indeed, if $X \prec Y$ and $X \neq Y, Y \not\subseteq A$ then $M \subset Y^\infty \subset A^\infty$ which is a contradiction.

To prove the converse, we suppose that X is an indecomposable strict-maximal code and M' is a freeable ∞-submonoid such that $M \subseteq M' \subseteq A^\infty$. This yields $X \subseteq X^\infty = M \subseteq M' = X'^\infty$ with X' being a strict code generating M'. Thus $X \prec X''$ for some subset $X'' \subseteq X'$. We show that X'' is also a strict-maximal code. If it is not so, then $X'' \cup x$ is a strict code for some x in $A^\infty - X''$. Since X is strict-maximal, $X \cup x (x \notin X$, because $x \notin (X'')^\infty \supseteq X)$ is not a strict code, we have then two different ∞- factorisations $(x_1, x_2, \ldots)$ and $(y_1, y_2, \ldots)$ over $X \cup x$ of some word of A^∞. Since every x_i, y_i that differ from x are in $X \subseteq X''^\infty$ they admit then ∞-factorizations over X''. Now, we replace every entry $x_i \neq x$ and $y_i \neq x$ in $(x_1, x_2, \ldots)$ and $(y_1, y_2, \ldots)$ with their ∞-factorizations over X'' and as a result we obtain two ∞-factorizations over $X'' \cup x$ of the same word. Since $X'' \cup x$ is a strict code, they must be identical, from which it follows that either $x \in X''$ or X is not a strict code. This contradiction shows that X'' must be strict-maximal, therefore $X'' = X'$ and by indecomposability of X we have either $X' = X$ or $X' \subseteq A$. Hence $X' = X$ or $X' = A$, in other words, $M' = M$ or $M' = A^\infty$. The proof is completed.

Example 2.8 *Consider the subset $M = \{w \in A^\infty : |w| \geq p\}, p$ is a prime number. It is clear that M is a freeable ∞-submonoid generated by the uniform (strict) code A^p which is strict-maximal and indecomposable. Therefore M is freeable-maximal by Theorem 2.7.*

3 Decomposition

In this section we study the relation $\prec$, namely, we are concerned with the question, does there exist for an arbitrary infinitary strict code X an indecomposable one by which X is written? Such a problem is not to be posed for finitary codes simply because the Zorn's Lemma guarantees this for them. In general, we do not know the answer to this question, but below we state some conditions under which a strict code can be written by an indecomposable one.

Theorem 3.1 *If X is a strict code with $X_{\rm fin}$ a finite finitary maximal code then X can be written by an indecomposable strict code.*

Proof.: First, we observe that for each strict code Y such that $X \prec Y$, we have $X_{\rm fin} \subseteq Y_{\rm fin}^*$ and it is not difficult to see that if $X_{\rm fin}$ is finite maximal code so is $Y_{\rm fin}$ and $X_{\rm fin} \prec Y_{\rm fin}$ (see [5], for example). Next, we denote $S(X) = \{Y \subseteq A^\infty : Y$ is a strict code : $X \prec Y\}$. Also, we denote $\|Y\| = \sum_{y \in Y_{\rm fin}} |y|$, so that for each $Y \in S(X)$ we have $\|Y\| < \infty$. Let N be the smallest value of $\|Y\|$ as Y runs through $S(X) : N = \min\{\|Y\| : Y \in S(X)\}$. We can see that if $Y_1 \prec Y_2$ with Y_1, Y_2 in $S(X)$ then $\|Y_1\| \geq \|Y_2\|$. Let Y be a strict code of $S(X)$ with $\|Y\| = N$. We show that Y is written by an indecomposable strict code and since $\prec$ is an order relation, so does X.

Consider $S(Y)$. Certainly $S(Y) \subseteq S(X)$. We use Zorn's Lemma to show that $S(Y)$ contains a maximal element which, therefore, is an indecomposable strict code and by which X can be written. For each $Z \in S(Y) \subseteq S(X)$, we have $X \prec Y \prec Z$ and $\|Z\| \leq \|Y\|$. As noted above $Y_{\rm fin} \prec Z_{\rm fin}$, and thus both $Z_{\rm fin}, Y_{\rm fin}$ are maximal codes and since $\|Y\|$ is of minimum value, it follows that $\|Z\| = \|Y\|$ and $Z_{\rm fin} = Y_{\rm fin}$.

Consider first an arbitrary countable chain in $S(Y) : Y_1 \prec Y_2 \prec \ldots$. We have $Y_{1{\rm fin}} = Y_{2{\rm fin}} = \ldots$ For each $s \geq 1$ and $x \in Y_{s{\rm inf}}, x$ does not belong to $Y^\omega_{s+1{\rm fin}}$, since $Y_{s+1{\rm fin}} = Y_{s{\rm fin}}$ and Y_s is a strict code. From $Y_s \prec Y_{s+1}$ it follows

$$x = x_{\rm fin}^{(s+1)} x_{\rm inf}^{(s+1)},$$

where $x_{\rm fin}^{(s+1)} \in Y^*_{s+1{\rm fin}}, x_{\rm inf}^{(s+1)} \in Y_{s+1{\rm inf}}$. By the same argument, we have

$$x_{\rm inf}^{(s+1)} = x_{\rm fin}^{(s+2)} x_{\rm inf}^{(s+2)},$$

where $x_{\rm fin}^{(s+2)} \in Y^*_{s+2{\rm fin}}, x_{\rm inf}^{(s+2)} \in Y_{s+2{\rm inf}}$, and so on. Clearly, there exist only finitely many i such that $x_{\rm fin}^{(s+1)} \neq \epsilon$, otherwise $x \in Y_{\rm fin}^\omega$. Thus there must be an integer $n(x, s)$ such that for a.e. $m \geq n(x, s)$

$$x_{\rm inf}^{(m)} = x_{\rm inf}^{(m+1)} = x(s).$$

Therefore $x(s) \in Y_{m{\rm inf}}$ for a.e. $m \geq n(x, s)$.

Now, consider the set P of A^∞ with $P_{\text{fin}} = Y_{1\text{fin}}, P_{\text{inf}} = \{x(s) : x \in Y_{s\text{inf}}, s = 1, 2, \ldots\}$. We verify that

(i) P is a strict code. In fact, if we have a relation, for example

$$x_1 \ldots x_m \alpha = y_1 \ldots y_n \beta$$

$(x_1, \ldots, x_m, \alpha) \in P_{(m+1)}, (y_1, \ldots, y_n, \beta) \in P_{(n+1)}$, then there exist $s, t \geq 1, x \in Y_{s\text{inf}}, y \in Y_{t\text{inf}}$ such that $\alpha = x(s), \beta = y(t)$. Let $l = \max\{n(x, s), n(y, t)\}$, then the words $x_1, \ldots x_m, \alpha, \beta, y_1, \ldots, y_n$ all are in Y_l and thus the above relation must be an identity. The other cases are treated similarly.

(ii) For all $i : Y_i \prec P$. As shown above, each $x \in Y_i$ is written in the form $x = x'x(i)$, where $x' \in Y^*_{1\text{fin}} = Y^*_{\text{fin}}$, so $x \in P^*_{\text{fin}} P_{\text{inf}}$. Thus $Y_i \subseteq P^\infty$ (moreover, $Y_i \subseteq P^*$). Further, for every $\alpha \in P_{\text{inf}}$, there is a positive integer s such that $x \in Y_{s\text{fin}}$ and $\alpha = x(s)$. If $n(x, s) \leq i$ then

$$\alpha = x(s) = x^{(n(x,s))}_{\text{inf}} = x^{(i)}_{\text{inf}} \in Y_{i\text{inf}}.$$

If $n(x, s) > i$, since $Y_i \prec Y_{n(x,s)}$ it follows $\alpha = x(s) \in Y_{n(x,s)}$ is present in the expression of some element of Y_i as a product of elements of $Y_{n(x,s)}$. Hence $Y_i \prec P$.

Thus we have proved that there exists an upper bound for any countable chain. Let now

$$Y_\alpha \prec Y_\beta \prec \ldots$$

be an uncountable chain (with cardinality at most continuum). We obviously can derive from this chain a countable subchain

$$Y_1 \prec Y_2 \prec \ldots$$

such that for each Y_γ from the uncoutable chain there exists Y_n from the countable subchain satisfying $Y_\gamma \prec Y_n$. For the latter one there exists a maximal element Y and it is easy to see that Y is also a maximal element for the uncountable chain. Now, in virtue of Zorn's Lemma Y is followed by a maximal element, i.e. Y is written by an indecomposable strict code. Proof is completed

In the next proposition we try to weaken the heavy demand of maximality of X_{fin}, but in compensation to this, the finiteness of X is required. We call X an *alphabetical code* if X is a subset of the alphabet A, otherwise we call it *nonalphabetical* one.

Theorem 3.2 *Each finite finitary nonalphabetical strict code is written by an indecomposable nonalphabetical strict code.*

Proof.: We note that for any finite finitary strict codes $X, Y : X \prec Y$ implies $\|X\| \geq \|Y\|$ as mentioned in the proof of the preceding theorem. The equality holds if and only if every word of Y occurs just once in the $*$-factorization of just one word of X, or equivalently, there exists a partition $Y = Y_1 \cup \ldots \cup Y_n (n \geq 1)$ such that for any $i : 1 \leq i \leq n$ there exists a word x (thus uniquely) such that x is a product of the words in Y_i (in some order). Hence if $\|X\| = \|Y\|$, we have $n = \text{Card } X \leq \text{Card } Y$ and in addition to this, if Card $X =$ Card Y then each Y_i is a singleton, which means $X = Y$.

We turn now to the proof. Let X be a finite finitary strict code. If X is indecomposable, we are done. Otherwise, we assume the contrary that X cannot be written by any indecomposable nonalphabetical strict code and as a consequence of this, we have an infinite chain of finite finitary codes: $X = X_0 \prec X_1 \prec X_2 \prec \ldots$, where $X_i \neq X_{i+1}$ and X_i is nonalphabetical for all $i = 1, 2, \ldots$ Certainly, we have: $\|X_0\| \geq \|X_1\| \geq \|X_2\| \geq \ldots$. Since $\|X_0\| < \infty$ it follows that for some integer $N : \|X_N\| = \|X_{N+1}\| = \ldots$. On the other hand, as we noted above: Card $X_N \leq$ Card $X_{N+1} \leq \ldots$. But for every $i \geq 1 : \|X_{N+1}\| \geq$ Card $X_{N+i} \min\{|x| : x \in X_{N+i}\} \geq$ Card X_{N+i}. Hence, there must be an integer M such that Card $X_{M+N} =$ Card X_{M+N+1}, therefore $X_{N+M} = X_{N+M+1}$ which is a contradiction with the assumpion that $X_i \neq X_{i+1}$ for all i. The proof is completed.

As a consequence of the preceding theorem, we shall have a decomposition theorem for finite finitary strict codes, but we recall some notations first. For more details one can consult [5] or [1]. Let X, Y be finitary codes over A and $X \prec Y$. Consider an alphabet B of the same cardinality as Y and a bijection $f : B \longrightarrow Y$. Because Y is a code, we can extend f to an isomorphism of B^* and Y^*, which we denote by the same $f, f : B^* \longrightarrow Y^*$. Let $Z = \{f^{-1}(x) : x \in X \subseteq Y^*\}$ and it is not difficult to see that Z is a code over B and it is a strict code if X and Y are strict codes. In this case we write $X = Y \bigotimes_B Z$. Conversely, if $Y \subseteq A^*, Z \subseteq B^*$ are codes (resp. strict codes) then the expression $X = Y \bigotimes_B Z$ stands for the following: there exists an isomorphism $f : B^* \longrightarrow Y^*$ such that $X = f(Z) \subseteq Y^*$. In this way X becomes a code (resp. strict code) and we have $X \prec Y$ if and only if B is the least alphabet such that $Z \subseteq B^*$. It is noteworthy that $\bigotimes$ is associative. Now we state our theorem.

Theorem 3.3 *Every finite finitary strict code X of A^* admits a finite decomposition:*

$$X = X_1 \bigotimes_B X_2 \bigotimes_C \ldots \bigotimes_D X_n,$$

where $X_1, X_2, \ldots X_n$ are indecomposable strict codes over the corresponding alphabets $A, B, \ldots, D$.

Proof.: The proof is proceeded by induction on $\|X\|$. If $\|X\| = 1$ then X is a letter, so it is indecomposable by definition. Suppose now that for every strict code X with $\|X\| < k$ the assertion is valid. Let $\|X\| = k$. If X is indecomposable, we are done; if not, by the preceding theorem, X is written by a nonalphabetical indecomposable strict code Y over $A^* : X \prec Y$. Thus, using the notations mentioned above, we have

$$X = Y \bigotimes_B Z.$$

Certainly, $|f^{-1}(x)| \leq |x|$ for each $x \in X$, therefore $\|Z\| \leq \|X\| = k$ and the equality holds if and only if $|f^{-1}(x)| = |x|$ for every x, i.e. when each word of Y is a letter meaning that $Y \subseteq A$ which is a contradiction. Thus we must have $\|Z\| < \|X\| = k$. By induction hypothesis Y admits a finite decomposition and so does X. Theorem is proved.

4 Theorem of Defect

In this concluding section we establish for strict codes a result, which is an analog of Theorem of Defect in the theory of finitary codes [2]. Note that Theorem of Defect was also proved for infinitary codes [4].

Theorem 4.1 *For any language X of A^∞, if X is not a strict code then X is written by a strict code of cardinality at most Card $X - 1$.*

Proof.: If Card $X = \infty$ nothing is to be done because X is always written by A or a subset of A and Card $A - 1 \leq \infty$. So we can assume Card $X < \infty$ and the proof is done by induction on Card X.

If Card $X = 1$ then X is a singleton strict code. Now suppose that for every X of cardinality not exceeding n the assertion is true. Let now X be a language of cardinality $n+1$ and X not be a strict code. We have then two different ∞-factorizations $(u_1, u_2, \ldots)$ and $(v_1, v_2, \ldots)$ over X with $u_1 \neq v_1$ of a word $\alpha \in A^\infty$. Further, we can suppose that $|v_1| > |u_1|$. Hence $v_1 = u_1\beta$ for some $\beta \in A^{+\infty}$. Consider two cases:

(i) If $v_1 \in A^N$, then we have $v_1 = u_1u_2\ldots$, therefore $\beta = u_2u_3\ldots$. Consider the language $X_1 = X - v_1$. If v_1 occurs among $u_2, u_3, \ldots$, say, $v_1 = u_k$ with k the smallest possible, $k > 1$; then we have

$$v_1 = u_1u_2\ldots u_{k-1}v_1$$

hence $v_1 = (u_1\ldots u_{k-1})^\omega \in X_1^\infty$.

If $v_1 \neq u_i$ for $i = 2, 3, \ldots$ then v_1 is obviuosly in X_1^∞. So we have $X \subseteq X_1^\infty$ and Card $X_1 <$ Card X. If X_1 is not a strict code, then by the induction hypothesis, X_1 is written by a strict code of cardinality $<$ Card$X_1 <$ Card$X$.

(ii) If $v_1 \in A^*$, then $\beta \in A^+$ and we put $X_1 = X - v_1 \cup \beta$. Clearly, $X \subseteq X_1^\infty$ and $\|X_1\| < \|X\|$. There are two possibilities

(ii.1) Replacing all the occurrences of v_1 by $u_1\beta$ in the equation:

$$u_1u_2\ldots = v_1v_2\ldots, \tag{2}$$

it becomes an identity. This means that $\beta = u_2 \in X - v_1$ and therefore Card$X_1 \leq$ Card$X - 1 = n$. The assertion follows by induction hypothesis applied to X_1.

(ii.2) If (2) does not become an identity after the replacement (ii.1), then we repeat the argument with X_1 until (i) or (ii.1) occurs. The process cannot go into infinity avoiding (i) or (ii.1) since $\|X_i\|, i = 1, 2, \ldots$ decreases strictly each time the argument is repeated. Thus we should obtain a finite sequence $X = X_0, X_1, \ldots X_s$ with $X_i \subset X_{i+1}^\infty$, Card $X_{i+1} \leq$ Card X_i for $i = 1, 2, \ldots, s-1$ and $X_s \subset C$, where C is some finite strict code with Card $C <$ Card X_s. So $X \subseteq C^\infty$ and Card $C \leq$ Card $X - 1$. The proof is complete.

Corollary 4.2 *Every two-element language is a strict code if and only if it is a code.*

Proof.: If $X = \{\alpha, \beta\}$ is not a code then X is not a strict code. Conversely, if X is not a strict code then by Theorem 4.1 $X \subseteq \{\gamma\}^\infty$ for some $\gamma \in A^\infty$. Since $\alpha \neq \beta, \gamma$ must belong to A^*. Hence X is not a code.

5 Acknowledgement

The authors express their sincere gratitude to the referee for his scrupulous work, especially, for his comments and suggestions, which helped them to improve the paper.

References

[1] Berstel J., Perrin D., "Theory of Codes," Academic Press, New York, 1985.

[2] Berstel J., Perrin D., Perrot J. F., Restivo A., *Sur la théorème du défaut,* Journal of Algebra **60** (1979), 169-180.

[3] Do Long Van, "Contribution to Combinatorics on Words," Thesis, Humboldt University, Berlin 1985.

[4] Do Long Van, *Languages écrits par un code infinitaire. Théorème du défaut,* Acta Cybernetica **7** (1986), 247-257.

[5] Lallement G., "Semigroups and Combinatorial Applications," John Wiley, New York, 1979.

[6] Nguyen Huong Lam, Do Long Van, *On a Class of Infinitary Codes,* Theoretical Informatics and Applications 24 (1990), 441-458.

[7] Staiger L., *On Infinitary Finite Length Codes,* Theoretical Informatics and Applications **20** (1986), 483-494.

(Received January 2, 1990)

Acta Cybernetica, Tom. 10, Fasc. 1-2, Szeged, 1991

Free submonoids and minimal ω-generators of R^ω

Igor Litovsky*

Abstract

Let A be an alphabet and let R be a language in A^+. An ω-generator of R^ω is a language G such that $G^\omega = R^\omega$. The language $\mathrm{Stab}(R^\omega) = \{u \in A^* : uR^\omega \subseteq R^\omega\}$ is a submonoid of A^*. We give results concerning the ω-generators for the case when $Stab(R^\omega)$ is a free submonoid which are not available in the general case. In particular, we prove that every ω-generator of R^ω contains at least one minimal ω-generator of R^ω. Furthermore these minimal ω-generators are codes. We also characterize the ω-languages having only finite languages as minimal ω-generators. Finally, we characterize the ω- languages ω-generated by finite prefix codes.

1 Introduction

Let A be an alphabet. Given a language R in A^*, the star operation provides a language, denoted by R^*, which is the smallest submonoid of A^* containing R. Conversely with each submonoid M of A^*, we can associate the family of languages G satisfying $G^* = M$, such languages are called $*$-generators of M. To obtain the most compact possible representation of M, one can seek the smallest $*$-generator of M if any with respect to inclusion. It is well known that, if M is submonoid of A^*, then the star root of M, that is the language $(M \setminus \{\varepsilon\}) \setminus (M \setminus \{\varepsilon\})(M \setminus \{\varepsilon\}))$ is the smallest $*$-generator of M [Br].

Here we consider the ω-power operation which for each language R in A^+, gives the language R^ω of infinite words $u_1 \dots u_n \dots$ where every u_n is a word in R. Conversely, with each language R^ω, we can associate a family of languages G satisfying $G^\omega = R^\omega$. Such languages are called ω- generators of R^ω. Note that for any ω- generator G of R^ω, the language $(G^2 \setminus G)$ is an ω-generator of R^ω, too. Hence the set of ω-generators does not have a minimum, therefore we consider its minimal elements. The question about the existence of minimal ω-generators remains to be solved in the general case. Here we approach the problem in a particular case in the following way. Each word u in A^* defines a left translation on A^ω. Given an ω-language L, the language $\mathrm{Stab}(L)$, already introduced in [St80], of words which stabilize L is a submonoid of A^*. For the case when $L = R^\omega$ and $\mathrm{Stab}(L)$ is a

*Laboratoire Bordelais de Recherche en Informatique, Univesité de Bordeaux I, Unité associé au C. N. R. S. n° 1304. Mailing address: 351, cours de la libération, F-33405 Talence Cedex FRANCE

free submonoid, we show that $\mathrm{Stab}(R^\omega)$ is of interest for the study of minimal ω-generators of R^ω. Previously other properties of the ω-languages whose stabilizer is free have been proved in [St80]. We establish here results which, for the general case, either do not hold (we propose counter-examples) or are not yet proved. The main result (Theorem 7) states that each ω- generator of R^ω contains at least one minimal ω-generator. Furthermore these minimal ω- generators are codes. Next we are interested in the finite, if any, minimal ω-generators of R^ω. By [LaTi] such ω-languages R^ω are closed sets with respect to the usual topology on A^ω. This makes us study the minimal ω-generators of closed ω-languages. We prove that they are right-complete sets (Theorem 9). Concerning the finite minimal ω-generators of R^ω, it is proved in [LaTi] and [Li] that one can decide, given a regular language R, whether $R^\omega = F^\omega$ for some finite set F. We also characterize the properties of all minimal ω-generators being finite languages (Theorem 15) and of only one ω-generator having the smallest possible cardinality (Theorem 17). Finally we show that the case of finite prefix codes is especially easy: some finite prefix code ω-generates R^ω if and only if some finite prefix code $*$-generates the stabilizer of R^ω and R^ω is a closed ω-language (Theorem 18). Unfortunately this result cannot be generalized for a larger class of codes.

Section 2 contains definitions and notation used in the following. In Section 3 we deal with the minimal ω-generators. The finite minimal ω-generators are the topic of Sections 4 and 5. Finally the finite prefix codes as ω-generators are investigated in the last section.

2 Preliminaries

Let A be a finite alphabet. We denote by A^* and A^ω the set of all finite words, and the set of all infinite words, respectively. Infinite words are called ω-words and subsets of A^* and A^ω are called languages and ω-languages, respectively. We denote by ε the empty word and by A^+ the language $A^* \setminus \{\varepsilon\}$. The concatenation is as usual extended to A^ω.

Let X be a language in A^* and let Y be a language or an ω-language. $X^{-1}Y$ stands for the language $\{v \in A^* \cup A^\omega : xv \in Y \text{ for some } x \in X\}$. X^* stands for the smallest submonoid of A^* with respect to inclusion, containing X and we denote by $\mathrm{Root}(X^*)$ the language $(X^* \setminus \{\varepsilon\}) \setminus (X^* \setminus \{\varepsilon\})(X^* \setminus \{\varepsilon\}))$.

Let u be a word and let v be word or an ω-word. The word u is a prefix of v if and only if $v \in u(A^* \cup A^\omega)$. Given a language X, $\mathrm{Pref}(X)$ is the language $\bigcup_{x\in X} \mathrm{Pref}\ (x)$.

Let u, v be two words. The word u is a suffix of v if and only if $v \in A^*u$. Given a language X, $\mathrm{Suff}(X)$ is the language $\bigcup_{x\in X} \mathrm{Suff}(x)$.

Let C be a language in A^*. C is a code if and only if each word has at most one factorization over C. A submonoid of A^* is free if and only if its root is a code [BePe]. C is an ifl-code [St86] if and only if each ω-word has at most one ω-factorization over C that is the equality $u_1 \ldots u_n \ldots = v_1 \ldots v_n \ldots$ where $u_n, v_n \in C$, implies that $u_n = v_n$ for all $n > 0$. C is a prefix code if and only if $CA^+ \cap C = \emptyset$. Note that every prefix code is an ifl-code and every ifl-code is a code. The converses do not hold [St86].

Let P be a subset of any monoid M, P is a right-complete set in M if and only if for each u in M, there exists v in M such that uv belongs to P^* [BePe].

Let X be a language in A^*, the adherence $\mathrm{Adh}(X)$ of X ([LinSt], [BoNi]) is the ω-language $\{w \in A^\omega : \mathrm{Pref}(w) \subseteq \mathrm{Pref}(X)\}$. Recall that $\mathrm{Adh}(X)$ is a closed set with respect to the usual topology on A^ω. Moreover L is a closed ω-language if and only if $L = \mathrm{Adh}(\mathrm{Pref}\ (L))$.

Let R be a language in A^+. R^ω is the ω- power of R, that is, the ω-language $\{u_1 \dots u_n \dots : u_n \in R\}$. We denote by $[R]_\omega$ the family $\{G \subseteq A^+ : G^\omega = R^\omega\}$. $G \in [R]_\omega$ is called an ω-generator of R^ω. The ω-language R^ω is said to be finitely ω-generated [LaTi] if and only if $R^\omega = F^\omega$ for some finite language F.

The stabilizer Stab(L) of an ω-language L is the language $\{u \in A^* : uL \subseteq L\}$ [St80].

3 Minimal ω-generators in the case when stab(R^ω) is a free submonoid

This work about the minimal ω-generators of R^ω is based on the stabilizer of R^ω. Recall first the following lemma.

Lemma 1 *[St80] [LiTi] Let L be a language. Then Stab(L) is a submonoid of A^*. Furthermore, in the case when $L = R^\omega$, Stab(R^ω) contains every ω-generator of R^ω.*

Lemma 2 *Let R be a language. Then $R^\omega = (R \setminus R(Stab(R^\omega) \setminus \{\varepsilon\}))^\omega$.*

Proof. Denote $R \setminus R(Stab(R^\omega) \setminus \{\varepsilon\})$ by G. The ω-language G^ω is contained in R^ω, since G is contained in R. Moreover, we have $R \subseteq (G \cup G\text{Stab}(R^\omega))$ and thus $R^\omega \subseteq (G \cup G\text{Stab}(R^\omega))R^\omega$. Now by definition of Stab(R^ω), it follows that $R^\omega \subseteq GR^\omega$ and finally $R^\omega \subseteq G^\omega$.

□

We now state a result concerning the subsets of free submonoids.

Lemma 3 *Let M be a free submonoid in A^* and G be a subset of M. Then the language $G \setminus G(M \setminus \{\varepsilon\})$ is a code.*

Proof. Denote $G \setminus G(M \setminus \{\varepsilon\})$ by G'. Let u be a word in G'^* and assume that $u \in g_1G'^* \cap g_2G'^*$ where g_1 and $g_2 \in G'$ and g_1 is a prefix of g_2. As $G' \subseteq M$, u has only one factorization in Root(M). Thus g_2 belongs to g_1M. Since $g_2 \in G'$, g_2 is equal to g_1.

□

In view of the above lemmas, we deduce:

Proposition 4 *Let R be a language such that Stab(R^ω) is a free submonoid in A^*. For each ω-generator G of R^ω, the language $G \setminus G(\text{Stab}(R^\omega) \setminus \{\varepsilon\})$ is a code ω-generating R^ω.*

We now give a characterization of codes which uses ω-words [LiSt].

Proposition 5 *Let C be a language in A^+. C is a code if and only if for each word u in C^+, the ω-word u^ω has a single ω-factorization over C.*

Proof. Assume that C is not a code. It follows that some u in C^+ has two different factorizations over C and hence u^ω has two different ω-factorizations over C. Assume now that for some u in C^+, u^ω has two different ω-factorizations over C. That is, $u^\omega = v_1 \dots v_n \dots$ where each $v_n \in C$ and the unique factorization of u in C^+ does not start with v_1. There exist four integers i, j, k and m such that $v_1 \dots v_k = u^i u'$ and $v_1 \dots v_m = u^{i+j} u'$ where u' is a prefix of u. It follows that

$u^{i+j}u'$ has two different factorizations over $C(v_1 \dots v_m$ and $u^j v_1 \dots v_k)$, that is C is not a code.

$\square$

So we can deduce a basic result for this paper.

Corollary 6 *Let C be a code in A^+. Then C is a minimal ω-generator of C^ω.*

Proof. Suppose we have a code C which is not a minimal ω-generator of C^ω. Then $(C \setminus \{v\})^\omega = C^\omega$ for some word $v \in C$. Hence $v^\omega \in (C \setminus \{v\})^\omega$ what implies that v^ω has two ω-factorizations over C. This contradicts the fact that C is a code.

$\square$

Hence the initial question about the existence of minimal ω-generators is answered.

Theorem 7 *Let R be a language such that $Stab(^\omega)$ is a free submonoid in A^*. Eaxh ω-generator G of R^ω contains at least one minimal ω-generaotr of R^ω. Furthermore, the code $G \setminus G(\mathrm{Stab}(R^\omega) \setminus \{\varepsilon\})$ is one of these.*

Without assuming that $\mathrm{Stab}(R^\omega)$ is free, the language $R \setminus R(\mathrm{Stab}(R^\omega) \setminus \{\varepsilon\})$ is genrally not a minimal ω-generator of R^ω, as shown by the following example.

Example 1 *Let R be the language $\{\varepsilon, b\}\{a\}\{b\}^*$. Here $Stab(R^\omega) = R^*$. but $R \setminus R(\mathrm{Stab}(R^\omega) \setminus \{\varepsilon\}) = R$ which is not a minimal ω-generator of R^ω, since $ab\ R^\omega$ is contained in $\{a, ab^2\}R^\omega$, which implies $(R \setminus \{ab\})^\omega = R^\omega$.*

We have actually proved that whenever $\mathrm{Stab}(R^\omega)$ is a free submonoid, then the minimal ω-generators of R^ω are exactly the codes ω-generating R^ω. However codes can ω-generate R^ω without $\mathrm{Stab}(R^\omega)$ being a free busmonoid, as shown below.

Example 2 *Let R be the language $\{aa, aaa, b\}$. Here $Stab(R^\omega) = R^*$ which is not a free submonoid. However the language $\{aa, aaab, b\}$ is a code ω-generating R^ω.*

4 The finite minimal ω-generators of R^ω

We have seen (Lemma 1) that $\mathrm{Stab}(R^\omega)$ contains every ω-generator of R^ω, but it is not necessarily an ω-generator of R^ω. As a counterexample consider $R = a^*b$ where $\mathrm{Stab}(R^\omega) = \{a, b\}^*$. However if R^ω is a closed subset of A^ω, we have the following result.

Lemma 8 *[LiTi]. Let R be a language such that R^ω is a closed subset in A^ω. Then $Stab(R^\omega)$ is the greatest ω-generator of R^ω.*

Now, in the case when R^ω is closed, we can link the notion of ω-generator of R^ω and the one of right-complete set in $\mathrm{Stab}(R^\omega)$.

Theorem 9 *Let R and G be two langauges such that R^ω as well as G^ω are closed ω-languages. Then the following two conditions are equivalent.*
(i) G is an ω-generator of R^ω
(ii) G is a right-complete set in $Stab(R^\omega)$.

Proof. Suppose G is an ω-generator of R. Les us recall [BePe] that G is a right-complete set in a submonoid M if and only if for each word u in M, there exists v in M satisfying $uv \in G^*$. Let u be a word in Stab(R^ω), we can wtite $u^\omega = g_1 \ldots g_n \ldots$ where each $g_n \in G$. Hence there exist two integers k, m and a prefix u' of u such that $k < m, u^k u'$ and $u^m u'$ belong to G^+. Moreover $u^m u' = u(u^{m-k-1}(u^k u'))$, thus uv belongs to G^+ where $v = u^{m-k-1}(u^k u')$ belongs to Stab(R^ω).

Conversely, if G is a right-complete set in Stab$(R^\omega), G^+ \subseteq$ Stab(R^ω) and Pref(Stab$(R^\omega)) \subseteq$ Pref(G^+). Hence Pref(Stab$(R^\omega)) =$ Pref(G^+). Moreover, Pref(Stab$(R^\omega)) =$ Pref$(R^\omega) =$ Pref(R^+). Now as G^ω and R^ω are closed ω-languages, $G^\omega =$ Adh(Pref(G^ω)) and $R^\omega =$ Adh(Pref(R^ω)). It follows that $G^\omega = R^\omega$.

□

Corollary 10 *Let R be a language such that R^ω is a closed ω-language and Stab(R^ω) is a free submonoid. Let G be a language such that G^ω is a closed ω-language. Then the following conditions are equivalent.*
(i) G is a minimal ω-generator of R^ω
(ii) G is a right-complete code in Stab(R^ω).

According to [LaTi], we know that if F is a finite language, F^ω is a closed ω-language. Then as a consequence of the above result we can characterize the finite minimal ω-generators of R^ω without using the ω-power.

Corollary 11 *Let R be a language such that R^ω is a closed ω-language and Stab(R^ω) is a free submonid. Then G is a finite minimal ω-generator of R^ω if and only if G is a finite right-complete code in Stab(R^ω).*

Remark. We cannot remove the assumption of R^ω being a closed ω-language. For example, with $R = a^*b$, Stab(R^ω) is the language $\{a,b\}^*$ and $\{a,b\}$ is a right-complete code in Stab(R^ω) but it is not an ω-generator or R^ω.

In [LaTi] and [Li] characterizations are given for R^ω being finitely ω-generated. In our current case we have the following characterization which does not hold in the general case [LaTi].

Theorem 12 *Let R be a language such that R^ω is a closed ω-language and Stab(R^ω) is a free submonid. R^ω is finitely ω-generated if and only if Root(Stab(R^ω)) is a finite language.*

Proof. Assume that Root(Stab(R^ω)) is an ifinite language and that G is a finite ω- generator of R^ω. As G is right-complete in Stab(R^ω), there exists a word $g \in G$ such that the set $E = \{u \in \text{Root(Stab}(R^\omega)) : \exists v \in \text{Stab}(R^\omega) \text{ with } uv \in gG^*\}$ is infinite. Since $G \subseteq$ Stab(R^ω), $g = g_1 \ldots g_k$ where each $g_i \in$ Root(Stab(R^ω)). Now since E is infinite, there exists $u_1 \in E$ such that $u_1 \neq g_1$. Then u_1 Stab$(R^\omega) \cap g_1$ Stab$R^\omega) \neq \emptyset$ given a contradiction.

□

However in the case when R^ω is finitely gnerated, some ω-generators could be infinite codes, as shown below.

Example 3 *Let R be the language $\{a^2, ba, ba^2\}$. Here Stab$R^\omega) = R^*$ and $\{a^2, ba\} \cup ba^2\{a^2\}^*\{ba, ba^2\}$ is an infinite code ω-generating R^ω.*

That leads to propose conditions for all minimal ω-generators of R^ω to be finite ones.

Lemma 13 *Let R be a language such that R^ω is a closed ω-language. If Root(Stab(R^ω)) is a finite ifl-code then all minimal ω-generators of R^ω are finite ifl-codes.*

Proof. Denote Root(Stab(R^ω)) by C. Assume that G is an infinite minimal ω-generator of R^ω. As C is a finite language, there exists a sequence (s_n) of C^* satisfying $s_0 = \varepsilon$ and for every integer $n, s_{n+1} = s_n r_{n+1}$ with $r_{n+1} \in C$ and $s_n C^+ \cap G$ is an infinite language. Moreover by Theorem 7, $G \cap GC^+ = \emptyset$. Hence for every integer n, s_n does not belong to G. As the ω-word $r_1 \dots r_n \dots$ belongs to C^ω, it is equal to $g_1 \dots g_n \dots$ where each $g_n \in G$. As C is an ifl-code. There exist $g \neq g'$ in G such that $gG^\omega \cap g'G^\omega \neq \emptyset$. Without loss of generality we may assume that g is a prefix of g'. Since C is an ifl-code, $g' \in gG^+$, this is a contradiciton with $G \cap GC^+ = \emptyset$.

□

The following lemma displays an important difference between regular codes and regular ifl-codes.

Lemma 14 *Let C be a regular code. If C is not an ifl-code then there exists an infinite code ω-generating C^ω.*

Proof. C being not an ifl-code, there exist words $\alpha, \beta \in C$ such that $\alpha \neq \beta$ and $\alpha C^\omega \cap \beta C^\omega \neq \emptyset$. Since C is regular, we deduce that $uv^\omega = u'v'^\omega$ for some $u \neq u'$ such that $u \in \alpha C^{i-1}, u' \in \beta C^{i-1}, v \in C^i$ and $v' \in C^i$. Moreover the language $uv^*(C^i \setminus \{v\}) \cup (C^i \setminus \{v\})$ is an infinite ω-generator of R^ω, which is a code since C^i is a code.

□

Noting that a finite language is a regular language and according to Lemmas 13 and 14, we state.

Theorem 15 *Let R be a language such that Stab(R^ω) is a free submonoid. All minimal ω-generators of R^ω are finite languages if and only if R^ω is a closed ω-language and Root(Stab(R^ω)) is a finite ifl-code.*

Remark. As shown by the following example, we cannot remove the assumption that Stab(R^ω) is a free subonoid.

Example 4 *Let R be the language $\{\varepsilon, b\}\{a, ab\}^*$. R is not a code, Stab(R^ω) $= R^*$ and Root(Stab(R^ω)) $= R$. However, by using the fact that Pref(R^+))$\cap$ Suff (R^+) $=$ $R^* \cup \{b\}$, we can prove that all minimal ω-generators of R^ω are finite langauges.*

As a consequence of Theorem 15, we characterize the minimal ω-generators of the whole language A^ω.

Corollary 16 *Let A be a finite alphabet. A language G is a minimal ω-enerator of A^ω if and only if G is a finite maximal prefix code in A^*.*

5 Uniqueness of the ω-generator of smallest cardinality

When R^ω is finitely ω-generated, there is obviously a smallest integer that can be the cardinality of some ω-generator of R^ω. But several ω-generators can have that integer for cardinality. For example, consider $R = \{aa, aaa, b\}$ where $\{aa, aaab, b\}$ is also an ω-generator of smallest cardinality. Here we seek languages R^ω such that only one ω- generator is of smallest cardinality.

Theorem 17 *Let R be a language such that R^ω is a closed ω-language and Stab(R^ω) is a free submonoid. Then the following conditions are equivalent.*
(i) Root(Stab(R^ω)) is the single ω-generator of smallest cardinality for R^ω
(ii) $2 \leq$ Card(Root(Stab(R^ω))) $< \infty$.

Proof. Denote Root(Stab(R^ω)) by C. If Card$(C) = 1$, then of course there are infinitely many ω-generators of cardinality 1. If C is infinite, then in view of Theorem 12, R^ω is not finitely ω- generated and all ω-generators are infinite languages.

Conversely, suppose $G \neq C$ is an ω-generator of smallest cardinality for R^ω. Let $g = cu$ be a word of G factorised by $c \in C$ and $u \in C^+$ (g exits since $G \neq C$). The language $(G \setminus \{g\}) \cup \{c\}$ is an ω-generator of smallest cardinality for R^ω. Step by step we obtain an ω-generator such as $(C \setminus \{c\}) \cup \{cu\}$ where $c \in C$ and $u \in C^+$. By factorizing u in $c'u'$, we can easily verify that $(C \setminus \{c\}) \cup \{cc'\}$ is an ω-generator of R^ω. Hence $(C \setminus \{c\})C \cup \{cc'\}$ is an ω-generator of R^ω, properly contained in C^2 : a contradiction since C^2 is a code and consequently C^2 is a minimal ω-generator of R^ω.

□

6 Case of finite prefix codes

In Section 3 we have seen that the language Stab(R^ω) does not allow us to characterize the languages R^ω ω-generated by a code. However for the finite prefix codes we have the following result.

Theorem 18 *Let R be a language. Then the following conditions ae équivalent.*
(i) $R^\omega = P^\omega$ for some finite prefix code P.
(ii) R^ω is a closed ω-language and Stab$(R^\omega) = P^*$ for some finite prefix code P.

Proof. If R^ω is a closed ω- language and Stab$(R^\omega) = P^*$ ffor some finite prefix code P, then $R^\omega = P^\omega$.

Conversely, let P be a finite prefix code such that $P^\omega = R^\omega$.

First $(^*)^{-1}$ Stab$(R^\omega) =$ Stab(R^ω). Indeed, let $uv \in$ Stab(R^ω) where $u \in P^*$. As $uvP^* \subseteq$ Pref(P^ω), for each z in P^*, there exists y in A^* such that $uvzy \in P^*$. P being a prefix code, $(P^*)^{-1}P^* = P^*$, hence $vzy \in P^* >$, that is $v \in$ Stab(R^ω).

Secondly (Stab$(R^\omega))^{-1}$ Stab$(R^\omega) \subseteq$ Stab(R^ω). Indeed, assume that $z \in$ (Stab$(R^\omega))^{-1}$ (Stab(R^ω)). Then Stab$(R^\omega) \cap$ (Stab$(R^\omega))z^{-1} \neq \emptyset$. Let u be a word in Stab$(R^\omega) \cap$ (Stab$(R^\omega))z^{-1}$ such that no any suffix of u is in Stab$(R^\omega) \cap$ (Stab$(R^\omega))z^{-1}$. As $u^\omega \in P^\omega$, there exist two words u_1, u_2 in A^* such that $u = u_1u_2$ and $u^iu_1 \in P^+$ and $u^{i+j}u_1 \in P^+$. Hence u_2, which is equal to $(u^iu_1)^{-1}u^{i+1}$, belongs to Stab(R^ω) according to the first point. Ditto u_2z belongs to Stab(R^ω), hence $u_2 \in$ Stab $(R^\omega) \cap$ (Stab$(R^\omega))z^{-1}$. It follows $u_2 = u$, next $u^i \in P^+$. Moreover $u^iz \in$ Stab(R^ω), hence $z \in$ Stab$R(^\omega)$. Finally (Pref(Stab$(R^\omega)))^* =$ Stab(R^ω). Indeed, let $u \in$ Stab$(R^\omega), u = cu'$ for some c in Pref(Stab(R^ω)). According to the second point, $u' \in$ Stab(R^ω) and step by step we obtain Stab$(R^\omega) \subseteq$ (Pref(Stab$(R^\omega)))^+$. This finishes the proof.

□

Finite prefix codes are particular finite ifl-codes. But R^ω cvan be ω-generated by a finite ifl-code without Stab(R^ω) being a free submonoid, as shown below.

Example 5 *Let R be the language $\{\varepsilon, b\}\{a, ab^2\}^*$. R is a finite ifl-code, hence R^ω is a closed o is a closed ω-language. However $Stab(R^\omega) = \{\varepsilon, b\}\{a, ab, ab^2\}^*$ and $Root(Stab(R^\omega)) = \{\varepsilon, b\}\{a, ab, ab^2\}$ which is not a code.*

When R^ω is ω-generated by an infinite prefix code, R^ω is never a closed ω-language and Stab(R^ω) is not necessarily an infinite prefix code.

Example 6 *Let R be the language $^*b$. R is an infinite prefix code, $Stab(R^\omega) = \{a, b\}^*$ which has $\{aa, b\}$ for root.*

Acknowledgments. The author is very deeply indebted to two referees for thorough reading of the first version of the manuscirpt. Their comments have resulted in a signifiant improvement in the exposition of the results.

References

[BePe] J. Berstel and D. Perrin, *Theory of codes;* Academic Press (1985).

[BoNi] L. Boasson and M. Nivat, *Adherences of languages;* Journal of Computer and System Sciences, 20 (1980) 285-309.

[Bu] J.R. Buchi, *On decision method in restricted second-order arithmetics;* Proc. Congr. Logic, Method. and Philos. Sci. (Stanford Univ. Press, 1962) 1-11.

[Br] J.A. Brzozowski, *Roots of Star Events;* J. ACM 14 (1967)3, 466-477.

[Do] Do Long Van, *Sur les ensembles gènèrateurs minimaux des sous-monoides de A^∞;* C.R. Acad. Sc. Paris; t. 300, Sèries I, n° 13, 1985.

[Ei] S. Eilenberg, *Automata, Languages and Machines;* Vol. A (Academic Press, New York, 1974).

[La] L.H. Landweber, *Decision problems for ω-automata,* Math. Syst. Theory 3 (1969) 376-384.

[LaTi] M. Latteux and E. Timmerman, *Finitely generated ω-languages;* Information Processing Letters 23 (1986) 171-175.

[LinSt] R. Lindner and L. Staiger, *Algebraische Codierungstheorie-Theorei der sequentiellen Codierungen;* Akademie-Verlag; Berlin 1977.

[Li] I. Litovsky, Rank of rational finitely generated ω-language; Proceedings of FCT'89, Lecture Notes in Computer Science 380, 308-317.

[LiSt] I. Litovsky and L. Staiger, *A characterization of codes via the ω-power,* (submitted paper).

[LiTi] I. Litovsky and E. Timmerman, *On generators of rational ω-power languages;* Theoretical Computer Science 53 (1987) 187-200.

[Ma] R. MacNaughton, *Testing and generating infinite sequences by a finite automaton;* Information and Control 9 (1966) 521-530.

[St80] L. Staiger, *A Note on Connected ω-languages;* EIK 16 (1980) 5/6, 245-251.

[St83] L. Staiger, *Finite-state ω-languages;* Journal of Computer and System Sciences, 27 (1983) 434-448.

[St86] L. Staiger, *On infinitary finite length codes;* Theoretical Informatics and Applications, vol. 20, n° 4 (1986) 486-494.

(Received December 18, 1990)

Acta Cybernetica, Tom. 10, Fasc. 1-2, Szeged, 1991

Special Families of Matrix Languages and Decidable Problems

A. Mateescu *

Abstract

We investigate some variants of simple matrix grammars. It is proved that the equivalence problem, the inclusion problem and other problems are decidables for this families of grammars. It would be noted that all these problems are undecidable for the family of simple matrix grammars.

1 Definitions and notations.

For an alphabet Σ we denote by Σ^* the free monoid generated by Σ under the operation of concatenation, and λ is the null element. The length of a string $\alpha \in \Sigma^*$ is denoted by $|\alpha|$. The set of natural numbers is denoted by N. If $n \in \mathrm{N}, n \geq 1$, then $[n]$ denotes the set $\{1, 2, \ldots n\}$. If $n \in N, n \geq 1$, and $\varphi : [n] \longrightarrow N$ is a function, then φ is a *n-function* and $|\varphi| = \Sigma_{i=1}^{n}\varphi(i)$. If $\theta = (\varphi, \psi)$, where φ and ψ are n-functions, then $|\theta| = |\varphi| + |\psi|$ and $\theta_i = \varphi(i) + \psi(i), i = 1, 2, \ldots, n$.

In order to obtain certain subfamilies of matrix languages we consider a special case of simple (linear, regular) matrix grammars, see [4] p. 68, definition 1.5.1.

Definition 1.1 *Let $n, k \in \mathrm{N}$ be such that $1 \leq k \leq n$ and let $\theta = (\varphi, \psi)$ be a pair of n-functions. A (n, k, θ)-linear matrix grammar (lmg) is a matrix grammar $G = (V, \Sigma, S, P)$ of degree n, where V is the nonterminal alphabet, Σ is the terminal alphabet, S is the start symbol ($S \notin V \cup \Sigma$) and P is a finite set of matrices of the following form:*

(i) $(S \longrightarrow A_1A_2 \ldots A_n)$, $A_i \in V$, $i = 1, \ldots, n$
(ii) $(A_1 \longrightarrow \alpha_1 B_1 \beta_1, \ldots, A_n \longrightarrow \alpha_n B_n \beta_n), A_i, B_i \in V, \alpha_i, \beta_i \in \Sigma^*, |\alpha_i| = \varphi(i), |\beta_i| = \psi(i)$, $i = 1, \ldots, n$
(iii) $(A_1 \longrightarrow \alpha_1, \ldots, A_n \longrightarrow \alpha_n), A_i \in V, i = 1, \ldots, n, \alpha_k \in \Sigma^*$, $0 \leq |\alpha_k| < |\theta|$ *and* $\alpha_i = \lambda$ *for* $i \neq k, i = 1, \ldots, n$.
A (n, k, θ)-linear matrix grammar is called *(n, k, φ)-regular matrix grammar (rmg)* iff $\psi(i) = 0, i = 1, \ldots, n$.

We define the direct derivation relation $\underset{G}{\Longrightarrow}$ and the derivation relation $\overset{*}{\underset{G}{\Longrightarrow}}$ as usually, see [3].

*University of Bucharest, Faculty of Mathematics, 14 Academiei, R-70109 Bucharest, ROMANIA

The language generated is:

$$L(G) = \{w | w \in \Sigma^*, S \overset{*}{\underset{G}{\Longrightarrow}} w\}$$

Definition 1.2 *The family of (n, k, θ)-linear matrix languages is:*

$$\mathcal{LM}_{n,k,\theta} = \{L | \exists G, (n, k, \theta) - \text{lmg and } L(G) = L\}$$

and the family of *(n, k, φ)-regular matrix languages is:*

$$\mathcal{RM}_{n,k,\varphi} = \{L | \exists G, (n, k, \varphi) - \text{rmg} \quad \text{and} \quad L(G) = L\}$$

Remark 1.3 *Let k be such that $1 \leq k \leq n$, let $\theta = (\varphi, \psi)$ be a pair of n-functions and let Σ be an alphabet. We consider the following two alphabets:*

$$\Sigma_1 = \{[\alpha] | \alpha \in \Sigma^* \text{ and } |\alpha| = |\theta|\} \quad \text{and} \quad \Sigma_2 = \{[\beta] | \beta \in \Sigma^* \text{ and } |\beta| < |\theta|\}.$$

For every $w \in \Sigma^*$ there exists and are unique two numbers $p, r \in \mathbf{N}$ such that

$$|w| = p|\theta| + r \quad \text{and} \quad 0 \leq r < |\theta|.$$

It is easy to remark that there exists a unique decomposition of w:

$$w = w_1 w_2 \ldots w_{k-1} u_k \beta v_k w_{k+1} \ldots w_n$$

such that for any $i = 1, \ldots, n, i \neq k, |w_i| = p\theta_i, |u_k| = p\varphi(k), |v_k| = p\psi(k)$, and $|\beta| = r$. Let w_k be the word $u_k v_k$. Then, there are the words $x_i^{(j)}, y_i^{(j)} \in \Sigma^*, j = 1, \ldots, p$ such that $|x_i^{(j)}| = \varphi(i), |y_i^{(j)}| = \psi(i)$ and $w_i = x_i^{(1)} x_i^{(2)} \ldots x_i^{(p)} y_i^{(p)} \ldots y_i^{(2)} y_i^{(1)}$, for all $i = 1, \ldots, n$. Let $z_i^{(j)}$ be the word $x_i^{(j)} y_i^{(j)}, i = 1, \ldots, n, j = 1, \ldots p$.

Using the above notations we shall define the function

$$\tau_\theta^{n,k} : \Sigma^* \longrightarrow \Sigma_1^* \Sigma_2$$

$$\tau_\theta^{n,k}(w) = [z_1^{(1)} z_2^{(1)} \ldots z_n^{(1)}][z_1^{(2)} z_2^{(2)} \ldots z_n^{(2)}] \ldots [z_1^{(p)} z_2^{(p)} \ldots z_n^{(p)}][\beta].$$

Note that for any θ, n and $k, \tau_\theta^{n,k}$ is a bijective function. Let us consider an example.

Example 1.4 *We choose $n = 3, k = 2, \theta = (\varphi, \psi)$ where $\varphi, \psi : [3] \rightarrow N, \varphi(1) = 1, \varphi(2) = 4, \varphi(3) = 1, \psi(1) = 2, \psi(2) = 1, \psi(3) = 3$ and let w be the word $a_1 a_2 \ldots a_{30}$. Note that $|\theta| = 12, |w| = 30$ and therefore $p = 2$ and $r = 6$. It results that:*

$$w = w_1 u_2 \beta v_2 w_2, \text{ where :}$$

$$w_1 = a_1 \ldots a_6, u_2 = a_7 \ldots a_{14}, \beta = a_{15} \ldots a_{20}, v_2 = a_{21} a_{22}, w_2 = a_{23} \ldots a_{30}.$$

Observe that:

$$z_1^{(1)} = a_1 a_5 a_6, z_1^{(2)} = a_2 a_3 a_4, z_2^{(1)} = a_7 \ldots a_{10} a_{22},$$

$$z_2^{(2)} = a_{11} \ldots a_{14} a_{21}, z_3^{(1)} = a_{23} a_{28} a_{29} a_{30}, z_3^{(2)} = a_{24} a_{25} a_{26} a_{27}.$$

From these remarks it follows that:

$$\tau_\theta^{n,k}(w) = [a_1 a_5 a_6 a_7 \ldots a_{10} a_{22} a_{23} a_{28} a_{29} a_{30}][a_2 a_3 a_4 a_{11} \ldots a_{14} a_{21} a_{24} a_{25} a_{26} a_{27}]$$

$$[a_{15} \ldots a_{20}].$$

2 Special properties and closure properties

The next two propositions prove the importance of the functions $\tau_\theta^{n,k}$.

Proposition 2.1 *If G is a (n,k,θ)-linear matrix grammar, then there is a regular grammar G' such that:*

$$L(G') = \tau_\theta^{n,k}\ (L(G)).$$

Proof. Let $G = (V, \Sigma, S, P)$ be a $(n,k,\theta)-$lmg and we define the regular Chomsky grammar, $G' = (V_N, V_T, S, P')$, see also the notations of remark 1.3

$$V_N = V^n \cup \{S\}, \quad V_T = \Sigma_1 \cup \Sigma_2$$

and the set of rules is:

$$\begin{array}{rl} P' = & \{S \longrightarrow (A_1,\ldots,A_n) | (S \longrightarrow A_1 \ldots A_n) \in P\} \cup \\ \cup & \{(A_1,\ldots,A_n) \longrightarrow [\alpha_1\beta_1 \ldots \alpha_n\beta_n](B_1,\ldots,B_n)| \\ & (A_1 \longrightarrow \alpha_1 B_1 \beta_1, \ldots, A_n \longrightarrow \alpha_n B_n \beta_n) \in P\} \cup \\ \cup & \{(A_1,\ldots,A_n) \longrightarrow [\alpha_k] | (A_1 \longrightarrow \alpha_1, \ldots, A_n \longrightarrow \alpha_n) \in P \text{ where} \\ & \alpha_i = \lambda, \text{ for any } i = 1,\ldots,n \text{ with } i \neq k\}. \end{array}$$

We can prove by induction on the length of the derivations, that:

$$A_1 A_2 \ldots A_n \overset{*}{\underset{G}{\Longrightarrow}} u_1 B_1 v_1 u_2 B_2 v_2 \ldots u_n B_n v_n$$

if and only if:

$$(A_1, A_2, \ldots A_n) \overset{*}{\underset{G'}{\Longrightarrow}} \tau_\theta^{n,k}\ (u_1 v_1 u_2 v_2 \ldots u_n v_n)(B_1, B_2, \ldots, B_n).$$

From the above equivalence and from the definition of $\tau_\theta^{n,k}$ it follows easily our proposition.

Now, we turn to the converse of proposition 2.1.

Proposition 2.2 *Let k be such that $1 \leq k \leq n$ and let $\theta = (\varphi, \psi)$ be a pair of n-functions. If the language $L, L \subseteq \Sigma_1^*\Sigma_2$, is a regular language, then there is a (n,k,θ)-linear matrix grammar, G, such that:*

$$L(G) = \eta_\theta^{n,k}(L),$$

where $\eta_\theta^{n,k}$ is the inverse function of $\tau_\theta^{n,k}$ (see also the remark 1.3).

Proof. Let $G' = (V_N, \Sigma_1 \cup \Sigma_2, S', P')$ be a regular grammar such that $L(G') = L$. Without loossing of generality, we can assume that the nonterminal rules of $P', A \longrightarrow \alpha B$, has the property that $\alpha \in \Sigma_1$ and also we can assume that the terminal rules of $P', A \longrightarrow \alpha$, has the property that $\alpha \in \Sigma_2$.

This assumption follows from the condition $L \subseteq \Sigma_1^*\Sigma_2$. We define the (n,k,θ)-linear matrix grammar $G, G = (V, \Sigma, S, P)$ where: $V = V_N \cup \{S\}$, with S a new symbol and the rules:

$$\begin{aligned} P \;=\; & \{(S \longrightarrow \underbrace{S' \ldots S'}_{n})\} \cup \{(A \longrightarrow \alpha_1 B \beta_1, \ldots, A \longrightarrow \alpha_n B \beta_n)| \\ & |A \longrightarrow [\alpha_1\beta_1 \ldots \alpha_n\beta_n]B \in P', |\alpha_i| = \varphi(i), |\beta_i| = \psi(i), i = 1, \ldots, n\} \cup \\ \cup\; & \{(A \longrightarrow \alpha_1, \ldots, A \longrightarrow \alpha_n)| A \longrightarrow \beta \in P', \beta \in \Sigma_2, \alpha_i = \lambda, \\ & i = 1, \ldots, n, i \neq k \text{ and } \alpha_k = \beta\} \end{aligned}$$

It follows easily that $L(G) = \eta_\theta^{n,k}(L)$.

Remark 2.3 *The propositions 2.1 and 2.2 are also true if G is a (n,k,φ) regular matrix grammar.*

Theorem 2.4 *For every $n, k \in \mathbf{N}$, with $1 \leq k \leq n$ and for every pair of n-functions, $\theta = (\varphi, \psi)$, the family $\mathcal{LM}_{n,k,\theta}$ is closed under union, intersection and complement.*

Proof. The closure under union is obvious. Therefore, it is enough to prove the closure under complement. If $L \in \mathcal{LM}_{n,k,\theta}, L \subseteq \Sigma^*$, then the language $\tau_\theta^{n,k}(L)$ is regular (Proposition 2.1). It follows that the language $L_1 = \Sigma_1^*\Sigma_2 - \tau_\theta^{n,k}(L)$ is also a regular language and $L_1 \subseteq \Sigma_1^*\Sigma_2$. From the proposition 2.2 we deduce that the language $\eta_\theta^{n,k}(L_1)$ is in $\mathcal{LM}_{n,k,\theta}$. But, $\tau_\theta^{n,k}$ is a bijective function and $\eta_\theta^{n,k}$ is the inverse function of $\tau_\theta^{n,k}$. It is easily to observe that $\eta_\theta^{n,k}(L_1) = \Sigma^* - L = CL$ and therefore $CL \in \mathcal{LM}_{n,k,\theta}$.

Corollary 2.5 *For every $n, k \in N$, with $1 \leq k \leq n$ and for every n-function φ, the family $\mathcal{RM}_{n,k,\theta}$ is closed under union, intersection and complement.*

3 Decidable problems

For a general discussion on decidable and undecidable problems in theory of matrix languages see the monography [3].

In the sequel we establish some decidable properties of the families $\mathcal{LM}_{n,k,\theta}$ and $\mathcal{RM}_{n,k,\varphi}$.

Theorem 3.1 *For every family $\mathcal{LM}_{n,k,\theta}$, the following problems are decidable:*

(1) Equivalence $(L_1 = L_2?)$

(2) Inclusion $(L_1 \subseteq L_2?)$

(3) Empty intersection $(L_1 \cap L_2 = \emptyset?)$

(4) Finite intersection (is $L_1 \cap L_2$ a finite set?)

(5) Empty complement $(CL = \emptyset?)$

(6) Finite complement (is CL a finite set?)

Proof.

(1) If $L_1, L_2 \in \mathcal{L}\mathcal{M}_{n,k,\theta}$, then the languages $L'_i = \tau_\theta^{n,k}(L_i), i = 1,2$ are regular languages (see proposition 2.1.). But $\tau_\theta^{n,k}$ is a bijective function and therefore $L_1 = L_2$ if and only if $L'_1 = L'_2$. The last equality is decidable.

(2) analogously.

(3)-(4). If $L_1, L_2 \in \mathcal{L}\mathcal{M}_{n,k,\theta}$, then $L_1 \cap L_2 \in \mathcal{L}\mathcal{M}_{n,k,\theta}$ (see theorem 2.4). But, for the family of simple matrix languages the emptiness problem and the finiteness problem are decidable problems (see [3]).

(5)-(6) If $L \in \mathcal{L}\mathcal{M}_{n,k,\theta}$, then $CL \in \mathcal{L}\mathcal{M}_{n,k,\theta}$ (see theorem 2.4) and the proof follows like in the (3)-(4) cases.

Corollary 3.2 *All problems from theorem 3.1 are decidable for every family* $\mathcal{R}\mathcal{M}_{n,k,\varphi}$.

Remark 3.3 *All problems from the theorem 3.1 are undecidable for whole family of simple linear (regular) matrix languages (see [3]).*

In what it follows we establish the relation between the families $\mathcal{L}\mathcal{M}_{n,k,\theta}$, $\mathcal{R}\mathcal{M}_{n,k,\varphi}$ and the Chomsky families of languages.

Obviously every family $\mathcal{L}\mathcal{M}_{n,k,\theta}$ is a proper subfamily of $\mathcal{L}\mathcal{M}$, the family of all simple matrix languages. It is well-known that $\mathcal{L}\mathcal{M}$ is a proper subfamily of $\mathcal{L}_1$, the family of dependent context languages (see [3], [4]). Therefore, for every $n, k, \in N, 1 \le k \le n$ and for every pair θ of n-functions is true that $\mathcal{L}\mathcal{M}_{n,k,\theta} \underset{\neq}{\subset} \mathcal{L}_1$.

Consequently, it follows that $\mathcal{R}\mathcal{M}_{n,k,\varphi} \underset{\neq}{\subset} \mathcal{L}_1$ for every $n, k, \in N, 1 \le k \le n$ and every n-function, φ.

Theorem 3.4 *(i) the regular family of languages, $\mathcal{L}_3$, is a proper subfamily of every family $\mathcal{R}\mathcal{M}_{n,k,\varphi}$ ($\mathcal{L}_3 \subsetneq \mathcal{R}\mathcal{M}_{n,k,\varphi}$).*

(ii) $\mathcal{L}_3$ is a proper subfamily of every family $\mathcal{L}\mathcal{M}_{n,k,\theta}$. ($\mathcal{L}_3 \underset{\neq}{\subset} \mathcal{L}\mathcal{M}_{n,k,\theta}$).

Proof. Let L be a regular language, $L \in \mathcal{L}_3$. There is a finite deterministic automaton, $A = (Q, \Sigma, \delta, q_1, F)$ such that $L(A) = L$.

We shall describe only the main constructions.

(i) We define a (n, k, φ)-regular matrix grammar, $G = (V, \Sigma, S, P)$, such that $L(G) = L$.

Let S be a new symbol and consider $V = Q \times Q \cup \{S\}$. The set of rules, P, is:

(1) $(S \longrightarrow (q_1, q_1)(q_2, q_2) \dots (q_n, q_n)), q_i \in Q,\ i = 1, \dots, n$, where q_1 is the initial state of A.

(2) $((p_1, r_1) \longrightarrow \alpha_1(p_1, t_1), (p_2, r_2) \longrightarrow \alpha_2(p_2, t_2), \dots, (p_n, r_n) \longrightarrow \alpha_n(p_n, t_n))$, for every $\alpha_i \in \Sigma^*$ such that $|\alpha_i| = \varphi(i), \delta(r_i, \alpha_i) = t_i$, and $p_i, r_i, t_i \in Q$ for $i = 1, \dots, n$.

(3) $((q_1, q_2) \longrightarrow \lambda, (q_2, q_3) \longrightarrow \lambda, \dots, (q_k, q'_k) \longrightarrow \beta, (q_{k+1}, q_{k+2}) \longrightarrow \lambda, \dots, (q_n, p) \longrightarrow \lambda)$, for every $\beta \in \Sigma^*$ such that $|\beta| < |\varphi|, \delta(q'_k, \beta) = q_{k+1}, p \in F$ and $q_i \in Q, i = 1, \dots, n, q'_k \in Q$.

One can prove that $L(G) = L$.

(ii) Analogously, we define a (n, k, θ)-linear matrix grammar, $G(V, \Sigma, S, P)$, such that $L(G) = L$.

Let V be the set $Q^4 \cup \{S\}$, where S is a new symbol. The rules in P are:

(1') $(S \longrightarrow (q_1, q_1, s_1, s_1)(q_2, q_2, s_2, s_2) \dots (q_n, q_n, s_n, s_n))$, for every $q_i, s_i \in Q, i = 1, \dots, n$ (q_1 is the initial state of $\mathcal{A}$)

(2') $((q_i, p_i, s_i, r_i) \longrightarrow \alpha_i(q_i, t_i, s_i, u_i)\beta_i), 1 \leq i \leq n$ for every $\alpha_i, \beta_i \in \Sigma^*$ such that $|\alpha_i| = \varphi(i), |\beta_i| = \psi(i), \delta(p_i, \alpha_i) = t_i, \delta(r_i, \beta_i) = u_i$, for every $q_i, p_i, s_i, r_i, t_i, u_i \in Q, i = 1, \dots, n$.

(3') $((q_1, p_1, p_1, r_1) \longrightarrow \lambda, (r_1, p_2, p_2, r_2) \longrightarrow \lambda, \dots, (r_{k-1}, p_k, t_k, r_k) \longrightarrow \beta, \dots, (r_{n-1}, p_n, p_n, r_n) \longrightarrow \lambda))$, for every $\beta \in \Sigma^*$ such that $|\beta| < |\theta|, \delta(p_k, \beta) = t_k, r_n \in F$ and $q_i, p_i, r_i \in Q, i = 1, \dots, n, t_k \in Q$.

It is not difficult to verify that $L(G) = L$.

Corollary 3.5 *For every family $\mathcal{LM}_{n,k,\theta}(\mathcal{RM}_{n,k,\varphi})$ the equivalence problem between an arbitrary language from the family and an arbitrary regular language is decidable.*

Proof. The proof uses theorem 3.4 theorem 3.1 (1) and corollary 3.2 (1).

Remark 3.6 *The above problem is also undecidable for the family of all linear (regular) simple matrix languages (see [3], [4]).*

For every $n \geq 2$, the family of context free languages, $\mathcal{L}_2$, is incomparable with any family $\mathcal{LM}_{n,k,\theta}$ or $\mathcal{RM}_{n,k,\varphi}$. This follows from the fact that the language $L = \{a^n b^n | n \geq 1\}^*$ is a context free language but L is neither a simple regular matrix language nor a simple linear matrix language (see [3]).

4 Further questions

For every families $\mathcal{LM}_{n,k,\theta}$ or $\mathcal{RM}_{n,k,\varphi}$ one can prove specifically pumping lemmas or other properties.

An interesting open problem arises from the following fact:

In the case $n = 1$ the family $\mathcal{LM}_{1,1,0}$ is the same with the family $\mathcal{L}_{i,j}$, see [1] and [2]. It is known, [5] and [6], that if $\mathcal{L}_{i,j}$ and $\mathcal{L}_{i',j'}$ are different families, then $\mathcal{L}_{i,j} \cap \mathcal{L}_{i',j'} = \mathcal{L}_3$.

From this remark in [5] and [6] it was found an important decidable problem.

For $n > 1$ this problem: "if the families $\mathcal{LM}_{n,k,\theta}$ and $\mathcal{LM}_{n,k,\theta}$, are different families, then $\mathcal{LM}_{n',k',\theta'} \cap \mathcal{LM}_{n,k,\theta}, = \mathcal{L}_3$" is an open problem. Analogously, this problem is open for the families $\mathcal{RM}_{n,k,\varphi}$.

References

[1] Amar, V., Putzolu, G. (1964), On a family of linear grammars, Information and Control 7, 283-291.

[2] Amar, V., Putzolu, G. (1965) Generalization of regular events, Information and Control 8, 56-63.

[3] Dassow, J., Paun, G. - Regulated Rewriting in Formal Language Theory, Akademie-Verlag, Berlin, 1989.

[4] Paun G., Gramatici matriciale, Ed. St. si Enc., Bucuresti, 1981.

[5] Semenov, A. L. (1974) Regularity of languages k-linear for various k, Soviet. Math. Dokl . 15, 492-436.

[6] Yehudai, A. (1980). The decidability of equivalence for a family of linear grammars, Information and Control, 47, 122-136.

(Received June 16, 1989)

Acta Cybernetica, Tom. 10, Fasc. 1-2, Szeged, 1991

Boolean-type retractable automata with traps

A. Nagy*

Abstract

As in other branches of the algebra, it is a natural idea to find connections between automata and their congruence lattices. For example, describe all automata whose congruence lattices are Boolean algebras. Although this problem will not be solved in this paper, we give a necessary condition for automata to be automata whose congruence lattices are Boolean algebras.

The main object of this paper is to describe a special class of automata with this (necessary) condition. More precisely, we describe all Boolean-type retractable automata (Definition 4.) with traps.

By an automaton we shall mean a system $\mathbf{A} = (A, X, \delta)$ consisting of a state set A, an input set X and a transition function $\delta : A \times X \longrightarrow A$ $(A \neq \emptyset, X \neq \emptyset)$.

Denote X^* the free monoid over X and e the empty word of X. The transition function δ can be extended to $A \times X^*$ such that

$$\delta(a,p) = \begin{cases} a & \text{if } p = e \\ \delta(\delta(a,q),x) & \text{if } p = qx \ (q \in X^*, x \in X) \end{cases}$$

for all $a \in A, p \in X^*$. As known, an equivalence relation α on the state set A is called a congruence on the automaton $\mathbf{A} = (A, X, \delta)$ if $(a, b) \in \alpha$ implies $(\delta(a, x), \delta(b, x)) \in \alpha$ for all $a, b \in A$ and $x \in X$. The set of all congurences of an automaton $\mathbf{A}$ forms a lattice. This lattice will be denoted by $\mathcal{L}(\mathbf{A})$. The least element and the greatest element of $\mathcal{L}(\mathbf{A})$ will be denoted by ι and ω, respectively.

If ρ is a congruence on an automaton $\mathbf{A} = (A, X, \delta)$ and A/ρ denotes the set of all ρ-classes $[a]\rho$ of $\mathbf{A}, a \in A$, then $\mathbf{A}/\rho = (A/\rho, X, \delta_\rho)$ is an automaton, where δ_ρ is defined by letting $\delta_\rho([a]\rho, x) = [\delta(a, x)]\rho$, for all $\alpha \in A$ and $x \in X$. The automaton $\mathbf{A}/\rho$ is called the factor automaton $\mathbf{A}$ modulo ρ.

If $\mathbf{R} = (R, X, \delta_R)$ is a subautomaton of an automaton $\mathbf{A} = (A, X, \delta)$ (here δ_R is the restriction of δ to R),then the subset R of A will be called a right ideal of $\mathbf{A}$ (see [2]). It can be easily verified that, for every right ideal R of $\mathbf{A}$,

$$\rho_R = \{(a, b) \in A \times A : a = b \quad \text{or} \quad a, b \in R\}$$

is a congruence on $\mathbf{A}$. This congurence is called the Rees congurence determined by R. The factor automaton $\mathbf{A}/\rho_R$ is called the Rees factor automaton of $\mathbf{A}$ modulo ρ_R (or modulo R).

A mapping φ of the state set A of an automaton $\mathbf{A} = (A, X, \alpha)$ into the state set B of an automaton $\mathbf{B} = (B, X, \beta)$ is called a homomrphism of $\mathbf{A}$ into $\mathbf{B}$ if $\lambda(\alpha(a, x)) = \beta(\lambda(a), x)$ for all $a \in A$ and $x \in X$. The congruence on $\mathbf{A}$ determined by the homomorphism λ will be denoted by con λ.

*Technical University of Budapest, Transport Engineering Faculty, Department of Mathematics, 1111 Budapest, XI., Müegyetem rkp. 9.

Definition 1 *A right ideal R of an automaton $\mathbf{A} = (A, X, \delta)$ will be called a retract right ideal if there is a homomorphism λ of $\mathbf{A}$ onto $\mathbf{R}$ which leaves the elements of R fixed. λ will be called a retract homomorphism of $\mathbf{A}$ onto $\mathbf{R}$.*

Definition 2 *We shall say that an automaton $\mathbf{A}$ is a retractable automaton if every right ideal of $\mathbf{A}$ is a retract right ideal.*

Theorem 3 *If $\mathbf{A}$ is an automaton such that $\mathcal{L}(\mathbf{A})$ is complemented [5], then $\mathbf{A}$ is a retractable automaton.*

Proof. Let R be a right ideal of an automaton $\mathbf{A} = (A, X, \delta)$. If $\mathcal{L}(\mathbf{A})$ is complemented, then, for the Rees congurence ρ_R, there is an element η_R in $\mathcal{L}(\mathbf{A})$ such that $\rho_R \wedge \eta_R = \iota$ and $\rho_R \vee \eta_R = \omega$. Then $\mathbf{A}/\eta_R = (A/\eta_R, X, \delta\eta_R)$ is isomorphic to $\mathbf{R} = (R, X, \delta_R)$.

Let λ_R denote the canonical homomorphism of $\mathbf{A}$ onto $\mathbf{A}/\eta_R$, that is $\eta_R =$ con λ_R. Identifying $\mathbf{A}/\eta_R$ with $\mathbf{R}$ it can be easily verfied that λ_R is a retract homomorphism of $\mathbf{A}$ onto $\mathbf{R}$.

Definition 4 *An automaton $\mathbf{A} = (A, X, \delta)$ will be called a Boolean- type retractable automaton if, for every right ideal R of $\mathbf{A}$, there is a retract homomorphism λ_R of $\mathbf{A}$ onto $\mathbf{R}$ such that $R \subseteq S$ implies con $\lambda_S \subseteq$ con λ_R in $\mathcal{L}(\mathbf{A})$, for all right ideals R and S of $\mathbf{A}$.*

Theorem 5 *If $\mathbf{A}$ is an automaton such that $\mathcal{L}(\mathbf{A})$ is a Boolean algebra, then $\mathbf{A}$ is a Boolean-type retractable automaton.*

Proof. Let $\mathbf{A} = (A, X, \delta)$ be an automaton such that $\mathcal{L}(\mathbf{A})$ is a Boolean algebra. As a Boolean algebra is a complemented lattice, it follows, by Theorem 3, that $\mathbf{A}$ is a retractable automaton. Let R and S be arbitrary right ideals of $\mathbf{A}$ with $R \subseteq S$. Then $\rho_R \subseteq \rho_S$, that is $\rho_R \wedge \rho_S = \rho_R$. From this equality it follows that $\eta_R \vee \eta_S = \eta_R$, that is $\eta_S \subseteq \eta_R$ which means that con $\lambda_S \subseteq$ con λ_R. Thus $\mathbf{A}$ is a Boolean-type retractable automaton.

Following [4], an element a_0 of the state set A is called a trap of the automaton $\mathbf{A} = (A, X, \delta)$ if $\delta(a_0, x) = a_0$, for all $x \in X$.

Theorem 6 *Every right ideal of a retractable automaton having traps contains a trap.*

Proof. Let R be a right ideal of a retractable automaton $\mathbf{A} = (A, X, \delta)$ with traps. Let a_0 be an arbitrary trap of $\mathbf{A}$ and λ_R a retract homomorphism of $\mathbf{A}$ onto $\mathbf{R}$. Then $\delta(\lambda_R(a_0), x) = \lambda_R(\delta(a_0, x)) = \lambda_R(a_0)$, for all $x \in X$. So $\lambda_R(a_0)$ is a trap of $\mathbf{A}$. As $\lambda_R(a_0) \in R$, the theorem is proved.

Definition 7 *An automaton will be called a one-trap-automaton (or an OT-automaton) if it has exactly one trap. If $\mathbf{A} = (A, X, \delta)$ is an OT-automaton with the trap a_0, then it will be denoted by $\mathbf{A} = (A, X, \delta; a_0)$.*

Theorem 8 *Every Boolean-type retractable automaton with traps has a homomorphic image which is a Boolean-type retractable OT-automaton.*

Proof. Let $\mathbf{A} = (A, X, \delta)$ be a Boolean-type retractable automaton with traps. Let R_t denote the set of all traps of $\mathbf{A}$. Then R_t is a right ideal of $\mathbf{A}$. It is evident that the factor automaton $\mathbf{A}/\rho_{R_t} = (A/\rho_{R_t}, X, \delta_{r_{R_t}})$ is an OT-autoamton. We show that $\mathbf{A}/\rho_{R_t}$ is also a Boolean-type retractable automaton. Let α denote the canonical homomorphism of $\mathbf{A}$ onto $\mathbf{A}/\rho_{R_t}$. Let R be an arbitrary right ideal of $\mathbf{A}/\rho_{R_t}$. Then $R\alpha^{-1} = \{a \in A : \alpha(a) \in R\}$ is a right ideal of $\mathbf{A}$. By Theorem 6, $R\alpha^{-1} \cap R_t \neq \emptyset$. So R contains the trap of $\mathbf{A}/\rho_{R_t}$. As $\mathbf{A}$ is a Boolean-type retractable automaton, there is a retract homomorphism $\lambda_{R\alpha^{-1}}$ of $\mathbf{A}$ onto $R\alpha^{-1}$. We define a mapping λ_R of $\mathbf{A}/\rho_{R_t}$ onto $\mathbf{R}$ as follows

$$\lambda_R(\alpha(a)) = \alpha(\lambda_{R\alpha^{-1}}(a)),$$

for all $a \in A$. We show that λ_R is a homomorphism of $\mathbf{A}/\rho_{R_t}$ onto $\mathbf{R}$. Let $a \in A, x \in X$ be arbitrary elements. Then

$$\delta_{\rho_{R_t}}(\lambda_R(\alpha(a)), x) = \lambda_{\rho_{R_t}}(\alpha(\lambda_{R\alpha^{-1}}(a)), x) = \alpha(\delta(\lambda_{R\alpha^{-1}}(a), x))$$
$$= \alpha(\lambda_{R\alpha^{-1}}(\delta(a, x))) = \lambda_R(\alpha(\delta(a, x))) = \lambda_R(\delta_{\rho_{R_t}}(\alpha(a), x)).$$

So λ_R is a homomorhism of $\mathbf{A}/\rho_{R_t}$ onto $\mathbf{R}$. It is evident that λ_R leaves the elements of R fixed. So λ_R is a retract homomorphism of $\mathbf{A}/\rho_{R_t}$ onto $\mathbf{R}$. Next we show that $R_1 \subseteq R_2$ implies con $\lambda_{R_2} \subseteq$ con λ_{R_1} in $\mathcal{L}(\mathbf{A}/\rho_{R_t})$, for every right ideals R_1 and R_2 of $\mathbf{A}/\rho_{R_t}$. Let R_1 and R_2 be right ideals of $\mathbf{A}/\rho_{R_t}$ with $R_1 \subseteq R_2$. Then $R_1\alpha^{-1} \subseteq R_2\alpha^{-1}$ and con $\lambda_{R_2\alpha^{-1}} \subseteq$ con $\lambda_{R_1\alpha^{-1}}$. Let a, b be arbitrary elements of $\mathbf{A}$ with $(\alpha(a), \alpha(b)) \in$ con λ_{R_2}, that is $\lambda_{R_2}(\alpha(a)) = \lambda_{R_2}(\alpha(b))$. Then $\alpha(\lambda_{R_2\alpha^{-1}}(a)) = \alpha(\lambda_{R_2\alpha^{-1}}(b))$ and so $\lambda_{R_2\alpha^{-1}}(a), \lambda_{R_2\alpha^{-1}}(b) \in R_t$ or $\lambda_{R_2\alpha^{-1}}(a), \lambda_{R_2\alpha^{-1}}(b) \notin R_t$ and $\lambda_{R_2\alpha^{-1}}(a) = \lambda_{R_2\alpha^{-1}}(b)$. Assume $\lambda_{R_2\alpha^{-1}}(a), \lambda_{R_2\alpha^{-1}}(b) \in R_t \subseteq R_1\alpha^{-1} \subseteq R_2\alpha^{-1}$. Then $\emptyset \neq [a]$ con $\lambda_{R_2\alpha^{-1}} \cap R_t = [b]$ con $\lambda_{R_1\alpha^{-1}} \cap R_t$. As $[a]$ con $\lambda_{R_2\alpha^{-1}} \cap R_t \subseteq [a]$ con $\lambda_{R_1\alpha^{-1}} \cap R_t$ and $[b]$ con $\lambda_{R_1\alpha^{-1}} \cap R_t \subseteq [b]$ con $\lambda_{R_1\alpha^{-1}} \cap R_t$, we get $\lambda_{R_1\alpha^{-1}}(a) \in R_t$ and $\lambda_{R_1\alpha^{-1}}(b) \in R_t$. Then $\alpha(\lambda_{R_1\alpha^{-1}}(a)) = \alpha(\lambda_{R_1\alpha^{-1}}(b))$, that is $\lambda_{R_1}(\alpha(a)) = \lambda_{R_1}(\alpha(b))$. So $(\alpha(a), \alpha(b)) \in$ con λ_{R_1}. Assume $\lambda_{R_2\alpha^{-1}}(a), \lambda_{R_2\alpha^{-1}}(b) \notin R_t, \lambda_{R_2\alpha^{-1}}(a) = \lambda_{R_2\alpha^{-1}}(b)$. Then $(a, b) \in$ con $\lambda_{R_2\alpha^{-1}} \subseteq$ con $\lambda_{R_1\alpha^{-1}}$, that is $\lambda_{R_1\alpha^{-1}}(a) = \lambda_{R_1\alpha^{-1}}(b)$. So $\alpha(\lambda_{R_1\alpha^{-1}}(a)) = \alpha(\lambda_{R_1\alpha^{-1}}(b))$ that is $\lambda_{R_1}(\alpha(a)) = \lambda_{R_1}(\alpha(b))$. Thus $(\alpha(a), \alpha(b)) \in$ con λ_{R_1}. Consequently con $\lambda_{R_2} \subseteq$ con λ_{R_1}. So $\mathbf{A}/\rho_{R_t}$ is a Boolean-type retractable OT-automaton.

By Theorem 8, we can concetrate our attention to only Boolean-type retractable OT-automata.

By [1], if $\underline{a}$ is a state of an automaton $\mathbf{A} = (A, X, \delta)$, then the intersection of all right ideals of $\mathbf{A}$ containing $\underline{a}$ is called the principal right ideals of $\mathbf{A}$ generated by a. This right ideal will be denoted by $R(a)$. It can be easily verfied that $R(a) = \delta(a, X^*) = \{\delta(a, p) : p \in X^*\}$.

The relation $\mathcal{R}$ on an automaton $\mathbf{A} = (A, X, \delta)$ defined as follows

$$\mathcal{R} = \{(a, b) \in A \times A : R(a) = R(b)\}$$

is an equivalence relation on A. The $\mathcal{R}$-class of A containing the elements a of A will be denoted by R_a. Let $R(a) - R_a$ be denoted by $R[a]$.

Theorem 9 *If a is an arbitrary element of an OT-automaton $\mathbf{A}$, then $R[a]$ is either empty (if a is the trap of $\mathbf{A}$) or a right ideal of $\mathbf{A}$ (if a is not the trap of $\mathbf{A}$).*

Proof. See, for example, [1].

Let $\mathbb{A} = (A, X, \delta)$ be an automaton. The factor automata $\mathbb{R}(a)/\rho_{R[a]}$ will be called the principal r-factors of $\mathbb{A}$ and they will be denoted by $\mathbb{R}\{a\}$. The state set and the transition function of $\mathbb{R}\{a\}$ will be denoted by $R\{a\}$ and $\delta_{\mathbb{R}\{a\}}$, respectively.

Let T be a set with a partially ordering $\leq$ such that every two-element subset of T has a lower bound in T and every non-empty subset of T having an upper bound in T contains a greates element. Then T is a semilattice under multiplication "$\cdot$" by letting $a \cdot b (a, b \in T)$ be the (necessarily unique) greatest lower bound of a and b in T. Following [6], a semilattice which can be constructed as above is called a tree. It is easy to see that the ideals of a tree T are those non-empty subsets I of T for which $b \in I$ and $a \leq b$ together imply $a \in I$ for all $a, b \in T$. If I is an ideal of a tree T, then the mapping π of T onto I letting $\pi(a)$ be the greatest element in the set $\{x \in I : x \leq a\}$ is a retract homomprhism of T onto I (see [6]). So evry ideal of a tree is a retract ideal [6].

Theorem 10 *The set $Prf(\mathbb{A})$ of all principal r-factors of a retractable OT-automaton $\mathbb{A} = (A, X, \delta; a_0)$ is a tree with the least element $\mathbb{R}\{a_0\}$ under ordering $\leq$ defined as follows: $\mathbb{R}\{a\} \leq \mathbb{R}\{b\}$ if and only if $R(a) \subseteq R(b)$.*

Proof. Let $\mathbb{A} = (A, X, \delta; a_0)$ be a retractable OT-automaton. It is evident that $\leq$ is a partially ordering on $\mathrm{Prf}(\mathbb{A})$. Let $\{\mathbb{R}\{a_j\} : j \in J\}$ be a non-empty subset of $\mathrm{Prf}(\mathbb{A})$. Assume that $\mathbb{R}\{a_j\} \leq R\{a\}$ for some $a \in A$. We shall prove that there is an element j_0 in J such that $\mathbb{R}\{a_j\} \leq R\{a_{j_0}\}$ for all $j \in J$. By the assumption that $\mathbb{R}\{a_j\} \leq \mathbb{R}\{a\}$ for all $j \in J$, we have $R(a_j) \subseteq R(a)$ for all $j \in J$. As $\mathbb{A}$ is a retractable automaton, there is a retract homomorphism λ of $\mathbb{A}$ onto $\mathbb{B} = (\cup R(a_j), X, \delta)$. So $\lambda(a) \in \{\cup R(a_j) : j \in J\}$ that is $\lambda(a) \in R(a_{j_0})$ for some $j_0 \in J$. Thus $R(\lambda(a)) \subseteq R(a_{j_0})$. It can be easily verified that $R(a_j) \subseteq R(a)$ implies $R(\lambda(a_j)) \subseteq R(\lambda(a))$ for all $j \in J$. So $R(\lambda(a)) = R(a_{j_0})$ that is $\mathbb{R}\{a_{j_0}\}$ is the greatest element of $\{\mathbb{R}\{a_j\} : j \in J\}$. Let $\mathbb{R}\{a\}$ and $\mathbb{R}\{b\}$ be arbitrary elements in $\mathrm{Prf}(\mathbb{A})$. Let K denote the set of all principal r-factors $\mathbb{R}\{c\}$ of $\mathbb{A}$ for which $\mathbb{R}\{c\} \leq \mathbb{R}\{a\}$ and $\mathbb{R}\{c\} \subseteq \mathbb{R}\{b\}$. As a_0 is in every right ideal of $\mathbb{A}$, it follows that K is not empty. So $\mathbb{R}\{a\}$ and $\mathbb{R}\{b\}$ have a common lower bound. Consequently the set of all principal r-factors of $\mathbb{A}$ forms a tree under ordering $\leq$. It is evident that $\mathbb{R}\{a_0\}$ is the least element of $\mathrm{Prf}(\mathbb{A})$.

Definition 11 *We shall say that an OT-automaton $\mathbb{A} = (A, X, \delta; a_0)$ is trapped if $\delta(a, x) = a_0$ for all $a \in A$ and $x \in X$.*

We note that a trivial automaton (when the state set has only one element) is trapped.

Definition 12 *An OT-automaton $\mathbb{A} = (A, X, \delta; a_0)$ will be called an r-simple OT-automaton if it is not rapped and $R = A$ or $R = \{a_0\}$, for all right ideals R of $\mathbb{A}$.*

Theorem 13 *Every principal r-factor of an OT-automaton is either r-simple or trapped.*

Proof. Let a be an arbitrary element of an OT-automaton $\mathbb{A} = (A, X, \delta; a_0)$. It is easy to see that $\mathbb{R}\{a\}$ is an OT-automaton. If $a = a_0$, then $\mathbb{R}\{a\}$ is trivial. Assume $a \neq a_0$. If $\delta(b, x) \in R[a]$ for all $b \in R_a$ and $x \in X$ such that $\delta(b, x) \notin R[a]$, then $\mathbb{R}\{a\}$ is r-simple.

Definition 14 *An OT-automaton is called an r-semisimple OT-automaton if its every principal r-factor is either trivial or r-simple.*

Next we characterize the r-semisimple OT-automata. Let $X^+ = X^* - \{e\}$, where e is the empty word.

Theorem 15 *An OT-automaton $\mathbb{A} = (A, X, \delta; a_0)$ is r-semisimple if and only if every right ideal R of $\mathbb{A}$ satisfies the following:*
(i) for every $a \in R$ there are elements $b \in R$ and $p \in X^+$ such that $a = \delta(b, p)$.

Proof. Let $\mathbb{A} = (A, X, \delta; a_0)$ be an r-semisimple OT-automaton. Let R be a right ideal of $\mathbb{A}$. If $a \in R$, then $R(a) \subseteq R$ and $\mathbb{R}\{a\}$ is either trivial (if $a = a_0$) or r-simple. We may assume $a \neq a_0$. Then $\mathbb{R}\{a\}$ is r-simple. So there is an element b in $R(a) - R[a]$, such that $\delta(b, x) \notin R[a]$ for some $x \in X$. So $(R[a]) \cup \{\delta(b, p) : p \in X^+\} = R(a)$ which implies $a = \delta(b, p)$ for some $p \in X^+$.

Conversely, assum that an OT-automaton $\mathbb{A} = (A, X, \delta; a_0)$ satifies (i). We prove that $\mathbb{A}$ is r-semisimple. Let c be an arbitrary element of A. We may assume $c \neq a_0$. Then $\mathbb{R}\{c\}$ is a non-trivial OT-automaton. We must show that $\mathbb{R}\{c\}$ is not trapped. Let a be an arbitrary element of $R(c)$ with $a \neq a_0$. Then, applying condition (i) for $R = R(c)$, there are elements b in $R(c)$ and p in X^+ such that $a = \delta(b, p)$. So $\mathbb{R}\{c\}$ is not trapped. Consequently $\mathbb{A}$ is r–semisimple.

We remark that condition (i) can be exchanged by condition
(ii) for every $a \in R$ there are elements $b \in R$ and $x \in X$ such that $a = \delta(b, x)$.

Definition 16 *Let $\mathbb{A} = (A, X, \delta_A)$ be a subautomaton of an automaton $\mathbb{B} = (B, X, \delta)$. We shall say that $\mathbb{B}$ is a dilation of $\mathbb{A}$ if there is a mapping φ of B onto A which leaves the elements of A fixed and $\delta(b, x) = \delta_A(\varphi(b), x)$ for all $b \in B$ and $x \in X$.*

Theorem 17 *An automaton is a Boolean-type retractable OT-automaton if and only if it is a dilation of an r-semisimple Boolean-type retractable OT-automaton.*

Proof. Assume that $\mathbb{B} = (B, X, \eta; a_0)$ is a Boolean-type retractable OT-automaton. Let $A = \eta(B, X) = \{\eta(b, x) : b \in B, x \in X\}$ and δ be the restriction of η to A. As $\mathbb{B}$ is a Boolean-type retractable automaton and A is a right ideal of $\mathbb{B}$, there is a retract homomorphism φ of $\mathbb{B}$ onto $\mathbb{A}$. Let $B_a = \{b \in B - A : \varphi(b) = a\}, a \in A$. If $b \in B_a$, then $A \ni \eta(b, x) = \varphi(\eta(b, x)) = \delta(\varphi(b), x)$. This implies that $\mathbb{B}$ is a dilation of $\mathbb{A}$.

It is evident that $\mathbb{A}$ is an r-semisimple OT-automaton (the r-semisimplicity follows from Theorem 15).

We show that $\mathbb{A}$ is a Boolean-type retractable automaton. Let R be an arbitrary right ideal of $\mathbb{A}$. Then R is also a right ideal of $\mathbb{B}$. So there is a retract homomorhism of $\mathbb{B}$ onto $\mathbb{R}$. The restriction of φ to $\mathbb{A}$ is a retract homomorphism of $\mathbb{A}$ onto $\mathbb{R}$. As $\mathbb{B}$ is a Boolean-type retractable automaton, it follows that $\mathbb{A}$ is a Boolean-type retractable one. Thus the first part of the theorem is proved.

Conversely, assume that an automaton $\mathbb{B} = (B, X, \eta)$ is a dilation of an r-semisimple Boolean-type retractable OT-automaton $\mathbb{A} = (A, X, \delta; a_0)$. Then there is a mapping φ of B into A which leaves the elements of A fixed and $\eta(b, x) = \delta(\varphi(b), x)$ for all $b \in B$ and $x \in X$.

It can be easily verfied that $\mathbb{B}$ is an OT-automaton with the trap a_0.

We prove that $\mathbb{B}$ is a Boolean-type retractable automaton. Let I be a right ideal of $\mathbb{B}$. Then $R = I \cap A$ is not empty and a right ideal of $\mathbb{A}$. As $\mathbb{A}$ is a Boolean-type retractable automaton, there is a retract homomorphism λ_R of $\mathbf{A}$ onto $\mathbf{R}$. Let $\wedge_I$ be defined on B as follows

$$\wedge_I(b) = \begin{cases} b & \text{if } b \in I \\ \lambda_R(\varphi(b)) & \text{if } b \notin I. \end{cases}$$

It is evident that $\wedge_I$ leaves the elements of I fixed and the restriction of $\wedge_I$ to A equals λ_R. We show that $\wedge_I$ is a homomorphism of $\mathbf{B}$ onto $\mathbf{I}$. Let $b \in B$ and $x \in X$ be arbitrary elements. If $b \in A$, then $\eta(\wedge_I(b), x) = \eta(\lambda_R(b), x) = \delta(\lambda_R(b), x) = \lambda_R(\delta(b, x)) = \wedge_I(\delta(b, x))$.

If $b \in (B - A) \cap I$, then $\eta(\wedge_I(b), x) = \eta(b, x) = \wedge_I(\eta(b, x))$, because $\eta(b, x) \in I$. If $b \in (B - A) - I$, then $\eta(\wedge_I(b), x) = \eta(\lambda_R(\varphi(b)), x) = \delta(\lambda_R(\varphi(b)), x) = \lambda_R(\delta(\varphi(b), x)) = \wedge_I(\delta(\varphi(b), x)) = \wedge_I(\eta(b, x))$.

Thus $\wedge_I$ is a retract homomorphism of $\mathbf{B}$ onto $\mathbf{I}$.

Assume that I and J are right ideals of $\mathbf{B}$ with $I \subseteq J$. Let $R_1 = I \cap A$ and $R_2 = J \cap A$. Then $R_1 \subseteq R_2$ and so con $\lambda_{R_2} \subseteq$ con λ_{R_1}. We show that con $\wedge_J \subseteq$ con $\wedge_I$. Assume $(a, b) \in$ con $\lambda_J; a, b \in B$. Then $\wedge_J(a) = \wedge_J(b)$. If $a, b \in J$, then, by the definition of $\wedge_J$, we have $a = b$. In this case $(a, b) \in$ con $\wedge_I$. If $a \in J$ and $b \notin J$, then $a = \wedge_J(a) = \wedge_J(b) = \lambda_{R_2}(\varphi(b)) \in A$. So $a \in A \cap J = R_2$ from which we get $a = \lambda_{R_2}(a)$. Thus $\lambda_{R_2}(a) = \lambda_{R_2}(\varphi(b))$ and so $\lambda_{R_1}(a) = \lambda_{R_1}(\varphi(b)) = \wedge_I(b)$. If $a \in I$, then $\lambda_{R_1}(a) = \wedge_I(a)$. If $a \notin I$, then, using $a = \varphi(a)$, we get $\lambda_{R_1}(a) = \lambda_{R_1}(\varphi(a)) = \wedge_I(a)$. So $\wedge_I(a) = \wedge_I(b)$. In the case $a \notin J$, the proof is similar.

If $a, b \notin J$, then $\wedge_J(a) = \wedge_J(b)$ implies $\lambda_{R_2}(\varphi(a)) = \lambda_{R_2}(\varphi(b))$. Then, by con $\lambda_{R_2} \subseteq$ con λ_{R_1}, we get $\lambda_{R_1}(\varphi(a)) = \lambda_{R_1}(\varphi(b))$. So $\wedge_I(a) = \wedge_I(b)$, because $a, b \notin I$.

Thus con $\wedge_J \subseteq$ con $\wedge_I$ has been proved. Consequently $\mathbf{B}$ is a Boolean-type retractable OT-automaton. Thus the theorem is proved.

Let $\mathbf{A} = (A, X, \delta; a_0)$ be an OT-automaton. Consider the set

$$A^0 = \begin{cases} A - \{a_0\} & \text{if } |A| > 1 \\ \{a_0\} & \text{if } A = \{a_0\} \end{cases}$$

and define the transition function $\delta^0 : A^0 \times X \longrightarrow A^0$ as follows

$$\delta^0(a, x) = \begin{cases} \delta(a, x) & \text{if } a, \delta(a, x) \in A^0 \\ \text{not defined} & \text{if } a \notin A^0 \text{ or } \delta(a, x) \notin A^0. \end{cases}$$

(A^0, X, δ^0) is a partial automaton which will be denoted by $\mathbf{A}^0$.

We note that if $\mathbf{A}$ is a trivial automaton then $\mathbf{A}^0$ equals $\mathbf{A}$.

A mapping φ of A_1^0 into A_2^0 will be called a partial homomorphism of a partial automaton $\mathbf{A}_1^0 = (A_1^0, X, \delta_1^0)$ into a partial automaton $\mathbf{A}_2^0 = (A_2^0, X, \delta_2^0)$ if $\delta_1(a, x) \in A_1^0$ implies $\delta_2(\varphi(a), x) \in A_2^0$ and $\delta_2(\varphi(a), x) = \varphi(\delta_1(a, x))$ for all $a \in A_1^0$ and $x \in X$.

Consider the following construction.

Let T be a tree with a least element ν. For every $\alpha \in T - \{\nu\}$, let $\mathbf{A}_\alpha = (A_\alpha, X, \delta_\alpha; a_\alpha)$ be an r-simple OT-automaton and let $\mathbf{A}_\nu = (\{a_0\}, X, \delta_\nu)$ be a trivial automaton.

Assume $A_\alpha \cap A_\beta = \emptyset$ if $\alpha \neq \beta$.

For all $\alpha, \beta \in T$ with $\alpha \geq \beta$, let $f_{\alpha,\beta}$ be a partial homomorphism of A_α^0 into A_β^0 such that

(i) $\varphi_{\alpha\alpha} = \mathrm{id}A^0_\alpha$ (the identical mapping of A^0_α),
(ii) $\varphi_{\beta,\gamma}(\varphi_{\alpha,\beta}(a)) = \varphi_{\alpha,\gamma}(a)$, for all $a \in A^0_\alpha$ and $\alpha \geq \beta \geq \gamma, (\alpha, \beta, \gamma \in T)$.

For every $a \in A^0_\alpha$ and $x \in X$, let $\overline{\alpha}[a, x]$ denote the greatest element of the set $\{\beta \in T : \delta_\beta(\varphi_{\alpha,\beta}(a), x) \in A^0_\beta\}$.

Assume that
(iii) for every $\alpha > \beta$ and $b \in A^0_\beta$ there are elements $a \in A^0_\alpha$,
$p = x_1x_2 \ldots x_n \in X^+ (x_1, x_2, \ldots x_n \in X)$ and
$\alpha_0, \alpha_1, \ldots, \alpha_n \in T$ such that $\alpha = \alpha_0 \geq \alpha_1 \geq \ldots \geq \alpha_n = \beta$ and

$$\begin{aligned}
&\overline{\alpha}_0[a, x_1] = \alpha_1,\\
&\overline{\alpha}_1[\delta_{\alpha_1}(\varphi_{\alpha,\alpha_1}(a), x_1), x_2] = \alpha_2\\
&\quad\vdots\\
&\overline{\alpha}_i[\delta_{\alpha_i}(\varphi_{\alpha,\alpha_i}(a), x_1x_2 \ldots x_i), x_{i+1}] = \alpha_{i+1},\\
&\quad\vdots\\
&\overline{\alpha}_{n-1}[\delta_{\alpha_{n-1}}((\varphi_{\alpha,\alpha_{n-1}}(a), x_1x_2 \ldots x_{n-1}), x_n] = \alpha_n,\\
&\delta_{\alpha_n}(\varphi_{\alpha,\alpha_n}(a), x_1x_2 \ldots x_n) = b.
\end{aligned}$$

Let $A = \{\cup A^0_\alpha : \alpha \in T\}$.

Define a transition function $\delta : A \times X \longrightarrow X$ as follows. If $a \in A^0_\alpha$ and $x \in X$, then let $\delta(a, x) = \delta_{\overline{\alpha}[a,x]}(\varphi_{\alpha,\overline{\alpha}[a,x]}(a), x)$.

It can be easily verified that $\mathbf{A} = (A, X, \delta; a_0)$ is an automaton which will be denoted by $[\mathbf{A}_\alpha, X, \delta_\alpha; \varphi_{\alpha,\beta}, T, a_0]$.

Next, we describe the r-semisimple Boolean-type retractable OT-automata.

Theorem 18 *An automaton is an r-semisimple Boolean-type retractable OT-automaton if and only if it is isomorphic with an automaton $[A_\alpha, X, \delta_\alpha; \varphi_{\alpha,\beta}, T, a_0]$ constructed as above.*

Proof. In the first part of the proof, we show that $\mathbf{A} = [A_\alpha, X, \delta_\alpha; \varphi_{\alpha,\beta}, T, a_0]$ is an r-semisimple Boolean-type retractable OT-automaton. It is evident that $\mathbf{A}$ is OT-automaton.

We show that $\mathbf{A}$ is retractable. Let I be a right ideal of $\mathbf{A}$. By the assumption that $\mathbf{A}_\alpha, \alpha \in T - \{\nu\}$ are r-simple and $\mathbf{A}_\nu$ is a trivial automaton, it follows that I is of the form $\{\cup A^0_\alpha : \alpha \in \Gamma\}$ where Γ is a non-empty subset of T. We show that Γ is an ideal of T. Let $\alpha \neq \nu$ be an arbitrary element of Γ. We show that $\beta < \alpha$ implies $\beta \in \Gamma$ for all $\beta \in T$. Let $\beta \in T$ and $b \in A^0_\beta$ be arbitrary elements such that $\beta < \alpha$. By (iii), there are elements a in $A^0_\alpha, p = x_1x_2 \ldots x_n$ in X^+ and $\alpha_1, \alpha_2, \ldots a_n$ in T with $\alpha \geq \alpha_1 \geq \ldots \geq \alpha_n = \beta$ such that

$$\begin{aligned}
&\delta(a, x_1) \in A^0_{\alpha_1}\\
&\delta(a, x_1x_2) = \delta(\delta(a, x_1), x_2) \in A^0_{\alpha_2}\\
&\quad\vdots\\
&\delta(a, x_1x_2 \ldots x_n) = b = \delta(\delta(\ldots \delta(\delta(a, x_1), x_2) \ldots), x_n) \in A^0_\beta.
\end{aligned}$$

As $a \in A^0_\alpha$ and $A^0_\alpha \subseteq I$, we have $\delta(a, x_1x_2 \ldots x_n) \in I$. So $A^0_\beta \cap I \neq \emptyset$ which implies $A^0_\beta \subseteq I$, that is $\beta \in \Gamma$. Thus Γ is an ideal of T.

Let π denote the retract homomorphism of T onto Γ. We define a retract homomorphism λ_I of $\mathbf{A}$ onto $\mathbf{I}$ as follows. For an arbitrary element a in A, let

$$\lambda_I(a) = \varphi_{\alpha,\pi(\alpha)}(a), \quad a \in A^0_\alpha. \tag{1}$$

By (i) and the fact that π is a retract homomorphism of T onto Γ, we can see that λ_I leaves the elements of I fixed. We prove that λ_I is a homomorphism. Let $a \in A$ and $x \in X$ be arbitrary elements. We may assume $a \neq a_0$. Let $a \in A^0_\alpha, \alpha \neq \nu$. Then

$$\lambda_I(\delta(a,x)) = \lambda_I(\delta_{\overline{\alpha}[a,x]}(\varphi_{\alpha,\overline{\alpha}[a,x]}(a),x)) =$$
$$= \varphi_{\overline{\alpha}[a,x],\pi(\overline{\alpha}[a,x])}(\delta_{\overline{\alpha}[a,x]}(\varphi_{\alpha,\overline{\alpha}[a,x]}(a),x)) =$$
$$= \delta_{\pi(\overline{\alpha}[a,x])}(\varphi_{\alpha,\pi(\overline{\alpha}[a,x])}(a),x) \in A^0_{\pi(\overline{\alpha}[a,x])}, \tag{2}$$

using (ii) and the fact that $\delta_{\overline{\alpha}[a,x]}\varphi_{\alpha,\overline{\alpha}[a,x]}(a),x) \in A^0_{\overline{\alpha}[a,x]}$ and so $\varphi_{\overline{\alpha}[a,x],\pi(\overline{\alpha}[a,x])}$ maps $\delta\overline{\alpha}[a,x](\varphi_{\alpha,\overline{\alpha}[a,x]}(a),x)$ into $A^0_{\pi(\overline{\alpha}[a,x])}$.

On the other hand, using (ii),

$$\delta(\lambda_I(a),x) = \delta(\varphi_{\alpha,\pi(\alpha)}(a),x) =$$
$$= \delta_{\overline{\pi(\alpha)}[\varphi_{\alpha,\pi(\alpha)}(a),x]}(\varphi_{\pi(\alpha),\overline{\pi(\alpha)}[\varphi_{\alpha,\pi(\alpha)}(a),x]}(\varphi_{\alpha,\pi(\alpha)}(a)),x)$$
$$= \delta_{\overline{\pi(\alpha)}[\varphi_{\alpha,\pi(\alpha)}(a),x]}(\varphi_{\alpha,\overline{\pi(\alpha)}[\varphi_{\alpha,\pi(\alpha)}(a),x]}(a),x) \in \tag{3}$$
$$\in A^0_{\overline{\pi(\alpha)}[\varphi_{\alpha,\pi(\alpha)}(a),x]}.$$

To prove that $\lambda_I(\delta(a,x)) = \delta(\lambda_I(a),x)$, we show that (2) and (3) are equal to each other.

First consider the case when $\overline{\alpha}[a,x] \geq \pi(\alpha)$. Then $\alpha \geq \overline{\alpha}[a,x] \geq \pi(\alpha)$, and so $\pi(\overline{\alpha}[a,x]) = \pi(\alpha)$. Thus (2) is equal to $\delta_{\pi(\alpha)}(\varphi_{\alpha,\pi(\alpha)}(a),x)$ which is in $A^0_{\pi(\alpha)}$. This also implies that $\overline{\pi(\alpha)}[\varphi_{\alpha,\pi(\alpha)}(a),x] = \pi(\alpha)$, because $\varphi_{\alpha,\pi(\alpha)}(a) \in A^0_{\pi(\alpha)}$ and $\delta_{\pi(\alpha)}(\varphi_{\alpha,\pi(\alpha)}(a),x)$ is not equal to the trap of $A_{\pi(\alpha)}$. Thus (3) is equal to $\delta_{\pi(\alpha)}(\varphi_{\alpha,\pi(\alpha)}(a),x)$ which means that (2) and (3) are equal to each other.

Consider the case $\overline{\alpha}[a,x] < \pi(\alpha)$. As Γ is an ideal of T and $\pi(\alpha) \in \Gamma$, we have $\overline{\alpha}[a,x] \in \Gamma$. So $\pi(\overline{\alpha}[a,x]) = \overline{\alpha}[a,x]$. Thus (2) is equal $\delta_{\overline{\alpha}[a,x]}(\delta_{\alpha,\overline{\alpha}[a,x]}(a),x)$. As $\varphi_{\pi(\alpha),\overline{\alpha}[a,x]}(\varphi_{\alpha,\pi(\alpha)}(a)) = \varphi_{\alpha\overline{\alpha}[a,x]}(a)$ (see (ii)), we have

$$\delta_{\overline{\alpha}[a,x]}(\varphi_{\pi(\alpha),\overline{\alpha}[a,x]}(\varphi_{\alpha,\pi(\alpha)}(a)),x) = \delta_{\overline{\alpha}[a,x]}(\varphi_{\alpha,\overline{\alpha}[a,x]}(a),x) \in A^0_{\overline{\alpha}[a,x]}.$$

So $\overline{\pi(\alpha)}[\varphi_{\alpha,\pi(\alpha)}(a),x] \geq \overline{\alpha}[a,x]$.

Let β be an arbitrary element of T with $\pi(\alpha) \geq \beta > \overline{\alpha}[a,x]$. Then $\delta_\beta(\varphi_{\pi(\alpha),\beta}(\varphi_{\alpha,\pi(\alpha)}(a)),x) = \delta_\beta(\varphi_{\alpha,\beta}(a),x)$. As $\beta > \overline{\alpha}[a,x]$, we get that $\delta_\beta(\varphi_{\alpha,\beta}(a),x)$ is the trap of A_β.

We note that this also implies that $\delta_{\pi(\alpha)}(\varphi_{\alpha,\pi(\alpha)}(a),x)$ is the trap of $A_{\pi(\alpha)}$, because $\delta_{\pi(\alpha)}(\varphi_{\alpha,\pi(\alpha)}(a),x) \in A^0_{\pi(\alpha)}$ would imply that $\delta_\beta(\varphi_{\pi(\alpha),\beta}$ $(\varphi_{\alpha,\pi(\alpha)}(a)),x) = \varphi_{\pi(\alpha),\beta}$ $(\delta_{\pi(\alpha)}(\varphi_{\alpha,\pi(\alpha)}(a),x) \in A^0_\beta$, contradicting that an automata $\delta_\beta(\varphi_{\alpha,\beta}(a),x) = \delta_\beta(\varphi_{\pi(\alpha),\beta}(\varphi_{\alpha,\pi(\alpha)}(a)),x)$ is the trap of A_β. Consequently $\overline{\pi(\alpha)}[\varphi_{\alpha,\pi(\alpha)}(a),x] \leq \overline{\alpha}[a,x]$. This and $\overline{\pi(\alpha)}[\varphi_{\alpha,\pi(\alpha)}(a),x] \geq \overline{\alpha}[a,x]$, proved above, together imply that $\overline{\pi(\alpha)}[\varphi_{\alpha,\pi(\alpha)}(a),x] = \overline{\alpha}[a,x]$. So (3) is equal to $\delta_{\overline{\alpha}[a,x]}(\varphi_{\alpha,\overline{\alpha}[a,x]}(a),x)$ which equals (2). Consequenlty λ_I is a homomorphism of **A** onto **I**.

To show that $\mathbb{A}$ is a Boolean-type retractable automaton, we prove that $I \subseteq J$ implies con $\lambda_J \subseteq$ con λ_I for all right ideals I and J of $\mathbb{A}$ where λ_I and λ_J constructed as in (1). Let $I \subseteq J$ be right ideals of $\mathbb{A}$. Then $I = \{\cup A^0_\alpha : \alpha \in \Gamma_I\}$ and $J = \{\cup A^0_\beta : \beta \in \Gamma_J\}$, Γ_I and Γ_J are ideals of T. Let π_I and π_J be the retract homomorphism of T onto I and J, respectively. Let λ_I and λ_J denote the retract homomorphism of $\mathbb{A}$ onto $\mathbb{A}/\rho_I$ and $\mathbb{A}/\rho_J$, respectivel. We must show that con $\lambda_J \subseteq$ con λ_I (that is $\lambda_J(a) = \lambda_J(b)$ implies $\lambda_I(a) = \lambda_I(b)$ for all $a, b \in A$). Let a and b be arbitrary elements in A with $a \in A^0_\alpha$ and $b \in A^0_\beta$, for some $\alpha, \beta \in T$. If $\lambda_J(a) = \lambda_J(b)$, then, by (1), $\varphi_{\alpha,\pi_J(\alpha)}(a) = \varphi_{\beta,\pi_J(\beta)}(b)$. So $\pi_J(\alpha) = \pi_J(\beta)$. As $I \subseteq J$, we get $\pi_I(\alpha) = \pi_I(\beta)$. As $\pi_J(\alpha) \geq \pi_I(\alpha)$ and $\pi_J(\beta) \geq \pi_I(\beta)$, we have $\varphi_{\alpha,\pi_I(\alpha)}(a) = \varphi_{\pi_J(\alpha),\pi_I(\alpha)}(\varphi_{\alpha,\pi_J(\alpha)}(a)) = \varphi_{\pi_J(\beta),\pi_I(\beta)}(\varphi_{\beta,\pi_J(\beta)}(b)) = \varphi_{\beta,\pi_I(\beta)}(b)$, that is $\lambda_I(a) = \lambda_I(b)$.

Consequently con $\lambda_J \subseteq$ con λ_I. Thus $\mathbb{A}$ is a Boolean-type retractable automaton.

We show that $\mathbb{A}$ is r-semisimple. Let I be a right ideal of $\mathbb{A}$. Then $I = \{\cup A^0_\alpha : \alpha \in \Gamma\}$ where Γ is an ideal of T. If $I = \{a_0\}$, then let $\lambda_I(a) = a_0$ for all $a \in A$. It is evident that λ_I is a homomorphism of $\mathbb{A}$ onto $\mathbb{I}$. Assume $I \neq \{a_0\}$. Let a be an arbitrary element of I. Then $a \in A^0_\alpha$ for some $\alpha \in \Gamma$. We show that there are elements b in I and p in X^+ such that $a = \delta(b,p)$. We may assume $a \neq a_0$. Let b be an arbitrary element in A^0_α. As A_α is r-simple, $R(b)$ (in A_α) equals A_α. So $a = \delta_\alpha(b,p)$ for some $p \in X^*$. If $|A^0_\alpha| > 1$ then b can be choosen such that $b \neq a$. In this case $p \in X^+$. If $A^0_\alpha = \{a\}$ then, by the r-simplicity of A, there is an element q in X^+ with $a = \delta_\alpha(a,q)$ (in the other case A_α must be trapped). Consequently $a = \delta(b,p)$ for some $b \in I$ and $p \in X^+$. Thus $\mathbb{A}$ is an r-semisimple automaton.

To prove the converse, let $\mathbb{A} = (A, X, \delta, a_0)$ be an r-semsimple Boolean-type retractable OT-automaton. Then there is a family Φ of retract homomorphisms φ_R of $\mathbb{A}$ onto $\mathbb{R}$, R are right ideals of $\mathbb{A}$, such that $R_1 \subseteq R_2$ implies con $\lambda_{R_2} \subseteq$ con λ_{R_1} for all right ideal R_1, R_2 of $\mathbb{A}$. It is evident that $A = \cup_{a\in A} R_a (= \cup_{a\in A} R^0\{a\})$.

By Theorem 10, the set Prf($\mathbb{A}$) of all principal r-factors of $\mathbb{A}$ is a tree under ordering $\leq$ defined as follows: $\mathbb{R}\{a\} \leq \mathbb{R}\{b\}$ if and only if $R(a) \subseteq R(b)$. The least element of Prf($\mathbb{A}$) is $\mathbb{R}\{a_0\}$, which is a trivial automaton. As $\mathbb{A}$ is r-semisimple, the automata $\mathbb{R}\{a\}, a \in A$ are r-simple OT-automata and $|\mathbb{R}\{a\}| = 1$ if and only if $a = a_0$. It is evident that $\mathbb{R}\{a\} \cap \mathbb{R}\{b\} = \emptyset$ if $\mathbb{R}\{a\} \neq \mathbb{R}\{b\}$. Let $\mathbb{R}\{a\}, \mathbb{R}\{b\}$ be arbitrary elements of Prf($\mathbb{A}$) with $\mathbb{R}\{a\} \geq \mathbb{R}\{b\}$ (that is $R(a) \supseteq R(b)$). Let $\varphi_{R(a),R(b)}$ denote the restriction of the retract homomorphism $\varphi_{R(b)} \in \Phi$ to $R(a)$. We show that $\varphi_{R(a),R(b)}$ maps R_a into R_b. Let z be an arbitrary element of R_a. If $b = a_0$, then $\varphi_{R(a),R(b)}(z) = a_0 \in R_b$. We show that $\varphi_{R(a),R(b)}(z) \in R_b$ also holds for all $b \neq a_0$. Assume, in an indirect way, that $\varphi_{R(a),R(b)}(z) \notin R_b$ for some $z \in R_a, b \neq a_0$. As $R[b]$ is a right ideal of $\mathbb{A}$, we get $R_b \not\ni \delta(\varphi_{R(a),R(b)}(z), x) = \varphi_{R(a),R(b)}(\delta(z,x))$ for all $x \in X$. As $\delta(z, X) = R(a)$, we get

$$\varphi_{R(a),R(b)}(R(a)) \subseteq R[b]. \tag{4}$$

As $\varphi_{R(b)}$ maps $\mathbb{A}$ onto $R(b)$ and leaves the elements of $R(b)$ fixed, we get that $\varphi_{R(a),R(b)}$ maps $R(a)$ onto $R(b)$ and leaves the elements of $R(b)$ fixed. Consequently

$$\varphi_{R(a),R(b)}(R(a)) = R(b),$$

contradicting (4). So $\varphi_{R(a),R(b)}(R_a) \subseteq R_b$. Thus $\varphi_{R(a),R(b)}$ determines a partial homomorphism $\varphi_{\mathbb{R}\{a\},\mathbb{R}\{b\}}$ of the partial automaton $\mathbb{R}^0\{a\}$ into the partial automaton $\mathbf{R}^0\{b\}$ as follows:

$$\varphi_{\mathbf{R}\{a\},\mathbf{R}\{b\}} : z \in R^0\{a\} \longrightarrow \varphi_{R(a),R(b)}(z).$$

We show that the family Φ^* of all partial homomorphisms $\varphi_{\mathbf{R}\{a\},\mathbf{R}\{b\}}$ ($\mathbf{R}\{a\}$, $\mathbf{R}\{b\} \in$ Prf ($\mathbf{A}$)) satisfies conditions (i), (ii) and (iii).

It is evident that $\varphi_{\mathbf{R}\{a\},\mathbf{R}\{b\}} = \mathrm{id}_{\mathbf{R}^0\{a\}}$ (see (i)).

To show (ii), let $\mathbf{R}\{a\} \geq \mathbf{R}\{b\} \geq \mathbf{R}\{c\}$ be arbitrary elements of Prf($\mathbf{A}$). Let e be an arbitrary element of R_a. As $\varphi_{R(a)}, \varphi_{R(b)}, \varphi_{R(c)} \in \Phi$, we have con $\varphi_{R(a)} \subseteq$ con $\varphi_{R(b)} \subseteq$ con $\varphi_{R(c)}$, that is $\varphi_{R(c)}(\varphi_{R(b)}(e)) = \varphi_{R(c)}(e)$. From this equality we get

$$\varphi_{R(b),R(c)}(\varphi_{R(a),R(b)}(e)) = \varphi_{R(a),R(c)}(e).$$

So the elements of Φ^* satisfy condition (ii).

To prove condition (iii), let $\mathbf{R}\{a\} > \mathbf{R}\{b\}$. Let $f \in R_b$. As $R(a) = \{\delta(a,p) : p \in X^*\}$, there is an element p in X^+ such that $f = \delta(a,p)$. If $p = x_1x_2\dots x_n$, then there are elements $a_1, a_2, \dots, a_n = b$ in A such that

$\delta(a, x_1) \in R_{a_1}$

$\delta(a, x_1x_2) \in R_{a_2}$

$\vdots$

$f = \delta(a, x_1x_2\dots x_n) \in R_b$.

The proof will be complete if we show that $\delta(a,x) = \delta_{\overline{\mathbf{R}\{a\}}[a,x]}(\varphi_{\mathbf{R}\{a\},\overline{\mathbf{R}\{a\}}[a,x]}(a), x), a \in \mathbf{R}_a$, where $\overline{\mathbf{R}\{a\}}[a,x]$ is the greatest element of the set $\{\mathbf{R}\{b\} \in \mathrm{Prf}(\mathbf{A}) : \delta_{\mathbf{R}\{b\}}(\varphi_{\mathbf{R}\{a\},\mathbf{R}\{b\}}(a), x) \in R^0\{b\}$. Let $a \in R_a$ and $x \in X$ be arbitrary elements. Then there is an element b in A such that $\mathbf{R}\{b\} \leq \mathbf{R}\{a\}$ and $\delta(a,x) \in R_b$. If c is an element of A such that $\mathbf{R}\{c\} \leq \mathbf{R}\{b\}$, we have $\delta(\varphi_{R(a),R(c)}(a), x) = \delta(\varphi_{R(b),R(c)}(\varphi_{R(a),R(b)}(a)), x) = \varphi_{R(b),R(c)}\delta(\varphi_{R(a),R(b)}(a), x) \in R_c$, because $\varphi_{R(b),R(c)}$ maps R_b into R_c.

If c is an element of A such that $\mathbf{R}\{c\} > \mathbf{R}\{b\}$, that is $R(c) \supset R(b)$, then $\delta(a,x) \notin R_c$ and so $\delta(a,x) = \varphi_{R(a),R(c)}\delta(a,x) = \varphi_{R(a),R(c)}(a), x) \notin R_c$. Consequently $\delta(\varphi_{R(a),R(c)}(a), x) \notin R_c$ for all $\mathbf{R}\{c\} > \mathbf{R}\{b\}$ and $\delta(\varphi_{R(a),R(d)}(a), x) \in R_d$ for all $\mathbf{R}\{d\} \leq \mathbf{R}\{b\}$. Thus $\mathbf{R}\{b\} = \overline{\mathbf{R}\{a\}}[a,x]$ and so $\delta(a,x) = \delta_{\overline{\mathbf{R}\{a\}}[a,x]}(\varphi_{\mathbf{R}\{a\},\overline{\mathbf{R}\{a\}}[a,x]}(a), x)$ for all $a \in R_a$ and $x \in X$. Then (iii) is satisfied and $\mathbf{A} \cong [\mathbf{R}\{a\}, X, \delta_{\mathbf{R}\{a\}}, \varphi_{\mathbf{R}\{a\},\mathbf{R}\{b\}}, \mathrm{Prf}(\mathbf{A}), a_0]$.

Thus the theorem is proved.

Example 1 *Let* $\mathbf{A} = (A, X, \delta)$ *be an automaton such that*

$$A = \{a_0, a_1, a_2, a_3, a_4\}, \quad X = \{x, y\}$$

and

δ	a_0	a_1	a_2	a_3	a_4
x	a_0	a_0	a_0	a_0	a_0
y	a_0	a_2	a_1	a_4	a_3

The right ideals of **A** are $I_0 = \{a_0\}$, $I_1 = \{a_0, a_1, a_2\}$, $I_2 = \{a_0, a_3, a_4\}$ and $I_3 = A$.

Consider the following mappings:

$$\begin{array}{llll}
\lambda_0 : A & \longrightarrow & \{a_0\} \text{ such that} & \lambda_0(a) = a_0 \text{ for all } a \in A, \\
\lambda_1 : A & \longrightarrow & I_1 \text{ such that} & \lambda_1(a) = a \text{ for all } a \in I_1 \text{ and} \\
 & & & \lambda_1(a_3) = a_1, \lambda_1(a_4) = a_2, \\
\lambda_2 : A & \longrightarrow & I_2 \text{ such that} & \lambda_2(a) = a \text{ for all } a \in I_2 \text{ and} \\
 & & & \lambda_2(a_1) = a_3, \lambda_2(a_2) = a_4, \\
\lambda_3 : A & \longrightarrow & A \text{ such that} & \lambda_3(a) = a \text{ for all } a \in A.
\end{array}$$

It can be easily verfied that λ_i is a retract homomorphism of **A** onto $\mathbf{I}_i, i = 0, 1, 2, 3$, and that **A** is an r-semisimple Boolean-type retractable OT-automaton (with the trap a_0).

Consider the following automata

$$\mathbf{A}_0 = (\{a_0\}, X, \delta_0), \quad \mathbf{A}_1 = (\{a_0, a_1, a_2\}, X, \delta_1), \quad \mathbf{A}_2 = (\{a_0, a_3, a_4\}, X, \delta_2),$$

where

δ_0	a_0
x	a_0
y	a_0

δ_1	a_0	a_1	a_2
x	a_0	a_0	a_0
y	a_0	a_1	a_1

δ_2	a_0	a_3	a_4
x	a_0	a_0	a_0
y	a_0	a_2	a_3

$\mathbf{A}_0$ is a trivial automaton, $\mathbf{A}_1$ and $\mathbf{A}_2$ are r-simple OT-automata. Let $T = \{0, 1, 2\}$, a subset of the set of the non-negative integers with the usual ordering. T is a tree. Let

$\varphi_{i,i}$ be the identical mapping of $A_i^0, i = 0, 1, 2$,

$\varphi_{1,0} : A_1^0 \longrightarrow \{a_0\}$ such that $\varphi_{1,0}(a) = a_0$ for all $a \in A_1^0$,

$\varphi_{2,0} : A_2^0 \longrightarrow \{a_0\}$ such that $\varphi_{2,0}(a) = a_0$ for all $a \in A_2^0$,

$\varphi_{2,1} : A_2^0 \longrightarrow A_1^0$ such that $\varphi_{2,1}(a_3) = a_1, \varphi_{2,1}(a_4) = a_2$.

It can be verified that $\varphi_{i,j}, i, j \in T$ with $i \geq j$, satisfy conditions (i) (ii) and (iii). Moreover

$$\mathbf{A} \cong [\mathbf{A}_i, X, \delta_i; \varphi_{i,j}, T, a_0].$$

References

[1] Babcsányi, I., Rees automaták (Hungarian with English summary), Matematikai Lapok, 29 (1977-81), No 1-3, 139-148.

[2] Babcsányi, I., Ideals in automata, Conference on System Theoretical Aspect in Computer Science, Held in Salgótarján/Hungary, May 24-26, 1982; Department of Math. Karl Marx Univ. of Economics, Budapest, 1982-2, 20-29.

[3] Gécseg, F. and I. Peák, Algebraic Theory of Automata, Akadémia Kiadó, Budapest, 1972

[4] Lex, W. and R. Wiegandt, Torsion theory for acts, Studia Sci. Math. Hung., 16 (1981), 263-280

[5] Szász, G., Introduction to Lattice Theory, Academic Press, New-York-London, 1963

[6] Tully, E.J., Semigroups in which each ideal is a retract, J. Austral Math. Soc. 9 (1969), 239-245

(Received January 18, 1990)

Acta Cybernetica, Tom. 10, Fasc. 1-2, Szeged, 1991

Language representations starting from fully initial languages

Gh. Paun S. Vicolov*

Abstract

It is proved that each regular/linear/context-free language is the image of a fully initial regular/linear/context-free language by an inverse homomorphism, as well as the intersection of two regular/linear/context-free fully initial languages, respectively. The converse of the latter assertion is not true for linear and for context-free languages.

1 Fully initial languages

For a context-free grammar $G = (V_N, V_T, S, P)$, one usually define the generated language as

$$L(G) = \{x \in V_T^* | S \overset{*}{\Longrightarrow} x\}$$

S. Horváth proposed to consider also the fully initial language generated by G, that is

$$L_{in}(G) = \{x \in V_T^* | A \overset{*}{\Longrightarrow} x, A \in V_N\}$$

(We denoted V^* the free monoid generated by V under the operation of concatenation; the null element of V^* is denoted by λ and $|x|$ denotes the length of $x \in V^*$. For $U \subseteq V$ and $x \in V^*$ we denote by $|x|_U$ the length of the string obtained by erasing from x all symbols not in U.)

We denote by REG, LIN, CF the families of regular, linear, context-free languages, respectively, and by FIREG, FILIN, FICF the families of fully initial languages generated by right-linear, linear and context-free grammars, respectively.

The fully initial languages were investigated in a series of papers [1], [3], [4], [6], [8]. In [3] it is proved that FICF is not closed under concatenation, intersection by regular sets and inverse homomorphisms; in fact, the proofs in [3] are true also for the family FILIN. The same nonclosure results hold also for the family FIREG (see [4]). On the other hand, FIX $\subset X$, strict inclusion, for each $X \in \{$REG, LIN, CF$\}$, [3], [4].

The above quoted results naturally raise the question of representing languages in a family X, X as above, starting from languages in FIX and using suitable operations. One such representation (characterization, in fact) has been done in [6], where it is proved

*University of Bucharest, Faculty of Mathematics, Str. Academiei 14, 70109 Bucuresti, ROMANIA

Theorem 1 *A language $L \subseteq V^*$ is in $X, X \in$ {REG, LIN, CF}, if and only if there is $L' \in FIX$ such that*

$$L = h_c(L' \cap \{c\}V^*)$$

where c is a new symbol and $h_c : (V \cup \{c\})^* \longrightarrow V^*$ is the homomorphism defined by $h_c(a) = a, a \in V, h_c(c) = \lambda$.

Two open problems are then raised in [6]:

(1) Can the homomorphism in above theorem be removed, that is suffices an intersection for obtaining a representation/characterization of X starting from languages in FIX?

(2) What about inverse homomorphism characterizations?

We affirmativelly solve here both these problems: there are such representations (sometimes characterizations).

In what follows, two languages will be considered eqnal if they differ only by the null string λ.

2 Characterization and representation results

Theorem 2 $X = \{h^{-1}(L) | L \in \text{FIX } h \text{ a homomorphism}\}$, $X \in$ {REG, LIN, CF}.

Proof. Each family X as above is closed under inverse homomorphisms [7], hence the inclusion $\supseteq$ is true.

Conversely, let $L \subseteq V^*$, be a context-free language. Denote by $d_a(L)$ the left derivative of L with respect to $a \in V$, that is

$$d_a(L) = \{x \in V^* | ax \in L\}.$$

We have

$$L = \bigcup_{a \in V} \{a\} d_a(L).$$

Each $d_a(L), a \in V$, is a context-free language; let $G_a = (V_{N,a}, V, S_a, P_a)$ be a λ-free grammar for $d_a(L)$. We construct the grammar

$$G = (V_N, V \cup \{c\}, S, P)$$

where c, S are new symbols,

$$V_N = \bigcup_{a \in V} V_{N,a} \cup \{S\}$$

and P contains the following rules:

(1) $S \longrightarrow ac$, if $a \in L, a \in V$,

(2) $S \longrightarrow aS_a, a \in V$,

(3) $A \longrightarrow x'$, for $A \longrightarrow x \in \bigcup_{a \in V} P_a$, and x' is obtained by replacing each terminal b in $x, b \in V$, by cb (the nonterminals in x remain unchanged),

(4) $A \longrightarrow x'$, for $A \longrightarrow x \in \bigcup_{a \in V} P_a$, and x' is obtained by replacing each terminal b in $x, b \in V$, by cb, excepting one occurrence of some $b \in V$ which is replaced by cbc (the nonterminals remain unchanged).

Let $L' = L_{in}(G)$ and consider the homomorphism $h : V^* \longrightarrow (V \cup \{c\})^*$ defined by $h(a) = ac, a \in V$. Clearly, $Im(h) = (V\{c\})^*$ and

$$L_{in}(G) = L(G) \cup \bigcup_{A \in V_N - \{S\}} L_A(G)$$

with

$$L_A(G = \{x \in V^* | A \overset{*}{\Longrightarrow} x \; in \; G\}$$

As each $x \in L_A(G), A \neq S$, is of the form $x = cy, y \in (V \cup \{c\})^*$, it follows that $Im(h) \cap L_A(G) = \emptyset$, hence $h^{-1}(L') = h^{-1}(L(G))$. On the other hand, we have

$$L(G) = L_1 \cup L_2 \cup L_3 \cup L_4$$

where

$$L_1 = L(G) \cap \{x \in (V \cup \{c\})^* || x|_c = |x|_V - 1\}$$

(the strings in L_1 are produced by using rules of the form (2) and (3), without using rules (1) and (4))

$$L_2 = L(G) \cap \{x \in (V \cup \{c\})^* || x|_c > |x|_V\}$$

(the strings in L_2 are obtained by using rules of forms (2), (3) and (4), namely at least two times rules of type (4))

$$L_3 = L(G) \cap \{x \in (V \cup \{c\})^* || x|_c = |x|_V, x = ya, a \in V, y \in (V \cup \{c\})^*\}$$

(the strings in L_3 are produced by using rules of types (2), (3) and (4), exactly one time a rule of type (4), but with cac not introduced on the rightmost position of the string).

$$L_4 = L(G) \cap (V\{c\})^*$$

(the strings in L_4 are produced by using rules of type (1), or of types (2), (3), (4), exactly one time a rule of type (4), with cac introduced on the rightmost position of the string)

Clearly, $Im(h) \cap (L_1 \cup L_2 \cup L_3) = \emptyset$, hence $h^{-1}(L(G)) = h^{-1}(L_4)$.

Moreover, $h(L) = L_4$ (from each derivation in $G_a, a \in V$, we can obtain a derivation in G and conversely, and $h(L) \subseteq Im(h) = (V\{c\})^*$), and h is an injective homomorphism, hence $h^{-1}(L_4) = L$, that is $L = h^{-1}(L_4) = h^{-1}(L(G)) = h^{-1}(L') = h^{-1}(L_{in}(G))$.

As one can see, if L is regular, then G is right-linear, and if L is linear, then G is linear too, which completes the proof.

Consider now the intersection. For LIN and CF we cannot obtain characterizations: consider the linear grammars

$$\begin{aligned} G_1 &= (\{S, A\}, \{a, b, c\}, S, \{S \longrightarrow Sc, S \longrightarrow A, A \longrightarrow aAb, \\ &\quad A \longrightarrow ab\}), \\ G_2 &= (\{S, A\}, \{a, b, c\}, S, \{S \longrightarrow aS, S \longrightarrow A, A \longrightarrow bAc, \\ &\quad A \longrightarrow bc\}). \end{aligned}$$

We have

$$\begin{aligned} L_{in}(G_1) &= \{a^n b^n c^m \mid n \geq 1, m \geq 0\} \\ L_{in}(G_2) &= \{a^n b^m c^m \mid n \geq 0, m \geq 1\} \end{aligned}$$

hence

$$L_{in}(G_1) \cap L_{in}(G_2) = \{a^n b^n c^n \mid n \geq 1\}$$

a language which is not context-free.

However, we can obtain representations of languages in X, $X \in$ {REG, LIN, CF}, as intersections of languages in FIX; as REG is closed under intersection, for this family we have in fact a characterization.

Theorem 3 *For each $L \in X$, $X \in$ {REG, LIN}, there are $L_1, L_2 \in$ FIX, such that $L = L_1 \cap L_2$.*

Proof. We consider here only the linear case; the regular case is a particular one.

Let $G = (V_N, V_T, S, P)$ be a linear grammar. Without loss of generality we may assume that each rule in P is of the next forms: $A \longrightarrow aB, A \longrightarrow Ba, A \longrightarrow a$ (for, each rule $A \longrightarrow a_1 \ldots a_n B b_m \ldots b_1$ can be replaced by $A \longrightarrow a_1 A_1, A_1 \longrightarrow a_2 A_2, \ldots, A_{n-1} \longrightarrow a_n C, C \longrightarrow C_1 b_1, C_1 \longrightarrow C_2 b_2, \ldots, C_{m-1} \longrightarrow B b_m$, etc.).

Consider the new symbols S_1, S_2 and construct the grammars

$$G_i = (V_N \cup \{S_i\}, V_T, S_i, P_i), i = 1, 2,$$

with

$$\begin{aligned} P_1 &= \{S_1 \longrightarrow x \mid x \in L(G), |x| \leq 1\} \\ &\quad \cup \{S_1 \longrightarrow xAy \mid S \stackrel{*}{\Longrightarrow} xAy \text{ in } G, |xy| \leq 2\} \\ &\quad \cup \{A \longrightarrow xBy \mid A \stackrel{*}{\Longrightarrow} xBy \text{ in } G, |xy| = 2\} \\ &\quad \cup \{A \longrightarrow a \mid A \longrightarrow a \in P, a \in V_T\}, \\ P_2 &= \{S_1 \longrightarrow x \mid x \in L(G), |x| \leq 2\} \\ &\quad \cup \{S_1 \longrightarrow xAy \mid S \stackrel{*}{\Longrightarrow} xAy \text{ in } G, |xy| \leq 2\} \\ &\quad \cup \{A \longrightarrow xBy \mid A \stackrel{*}{\Longrightarrow} xBy \text{ in } G, |xy| = 2\} \\ &\quad \cup \{A \longrightarrow ab \mid A \stackrel{*}{\Longrightarrow} ab \text{ in } G, a, b \in V_T\}. \end{aligned}$$

Clearly,

$$\begin{aligned} L(G_1) &= L(G_2) = L \\ L_{in}(G_1) &= L(G_1) \cup \bigcup_{A \neq S_1} L_A(G_1) \\ L_{in}(G_2) &= L(G_2) \cup \bigcup_{A \neq S_2} L_A(G_2) \end{aligned}$$

and

$$\begin{aligned} L_A(G_1) &\subseteq \{x \in V_T^* \mid |x| = 2k+1, k \geq 0\}, \ A \neq S_1, \\ L_A(G_2) &\subseteq \{x \in V_T^* \mid |x| = 2k, k \geq 1\}, \ A \neq S_2. \end{aligned}$$

Therefore,

$$L_{in}(G_1) \cap L_{in}(G_2) = L(G_1) \cap L(G_2) = L.$$

A similar representation theorem can be obtained also for the context-free case.

Theorem 4 *For each* $L \in$ CF, *there are* $L_1, L_2 \in$ FICF, *such that* $L = L_1 \cap L_2$.

Proof. Let $L \subseteq V^*$ be a context-free language and consider

$$\text{even}(L) = L \cap \{ab \mid a,\ b \in V\}^*,$$

$$\text{odd}(L) = L \cap \{ab \mid a,\ b \in V\}^* V.$$

Clearly, $L =$ even$(L) \cup$ odd(L) and even(L), odd(L) are context-free languages (CF is closed under intersection by regular sets).

On the other hand,

$$L = \bigcup_{a \in V} \{a\} d_a(L)$$

and

$$\text{even}(L) = \bigcup_{a \in V} \{a\} \ \text{odd}(d_a(L)),$$

$$\text{odd}(L) = \bigcup_{a \in V} \{a\} \ \text{even}(d_a(L)).$$

Therefore

$$\begin{aligned} L &= \text{even}(L) \cup \bigcup_{a \in V} \{a\} \ \text{even}(d_a(L)) \\ &= \text{odd}(L) \cup \bigcup_{a \in V} \{a\} \ \text{odd}(d_a(L)). \end{aligned}$$

All languages even$(d_a(L))$, odd$(d_a(L))$, $a \in V$, are context-free. In view of the super-normal form theorem in [2], [5], there are the grammars

(i) $G_1 = (V_{N,1}, V_T, S_1, P_1)$

$G_{a,1} = (V_{N,a,1}, V_T, S_{a,1}, P_{a,1})$

such that $L(G_1) =$ even(L), $L(G_{a,1}) =$ even$(d_a(L)), a \in V$, and the nonterminal rules in $P_1, P_{a,1}, a \in V$, are in the (2,0,0) normal form (of type $A \longrightarrow xBC$, $x \in V_T^*$, $|x| = 2, A, B, C$ nonterminals), whereas the terminal rules $A \longrightarrow w$ have $|w|$ in the length set of the generated language, that is $|w|$ is even;

(ii) $G_2 = (V_{N,2}, V_T, S_2, P_2)$

$G_{a,2} = (V_{N,a,2}, V_T, S_{a,2}, P_{a,2})$

such that $L(G_2) =$ odd(L), $L(G_{a,2}) =$ odd$(d_a(L))$, $a \in V$, and the nonterminal rules in $P_2, P_{a,2}, a \in V$, are in the (1,0,0) normal form (of type $A \longrightarrow bBC$, $b \in$

V, A, B, C nonterminals), whereas the terminal rules $A \longrightarrow w$ have $|w|$ in the length set of the generated language, that is $|w|$ is odd.

Now, it is easy to see that $L_{in}(G_1), L_{in}(G_{a,1})$, $a \in V$, contain only strings of even lengths, whereas $L_{in}(G_2), L_{in}(G_{a,2})$, $a \in V$, contain only strings of odd lengths (induction on the number of rules used in a derivation).

Assume all vocabularies $V_{N,i}, V_{N,a,i}, i = 1, 2$, pairwise disjoint and construct the grammars

$$G'_i = (V'_{N,i}, V_T, S'_i, P'_i),\ i = 1, 2,$$

with

$$\begin{aligned} V'_{N,i} &= V_{N,i} \cup \bigcup_{a\in V} V_{N,a,i} \cup \{S'_i\}, \\ P'_i &= P_i \cup \bigcup_{a\in V} P_{a,i} \cup \{S'_i \longrightarrow S_i\} \cup \{S'_i \longrightarrow aS_{a,i} | a \in V\}. \end{aligned}$$

From the above relations we have

$$\begin{aligned} L(G'_1) &= L(G'_2) = L, \\ L_{in}(G'_i) &= L_{in}(G_i) \cup \bigcup_{a\in V} \{a\} L_{in}(G_{a,i}),\ i = 1, 2 \end{aligned}$$

and, from the construction of G'_i, we obtain

(a) if $w \in L_{in}(G'_1) - L(G'_1)$, then $|w|$ is even,

(b) if $w \in L_{in}(G'_2) - L(G'_2)$, then $|w|$ is odd.

In conclusion, $L_{in}(G'_1) \cap L_{in}(G'_2) = L(G'_1) \cap L(G'_2) = L$, and the proof is over.

References

[1] T. Balanescu, M. Gheorghe, Gh. Paun, On fully initial grammars with regulated rewriting, *Acta Cybernetica*, 9, 2 (1989), 157-165.

[2] M. Blattner, S. Ginsburg, Canonical forms of context-free grammars and position restricted grammar forms, *Lect. Notes Computer Sci.*, 56, Springer-Verlag, 1979, 49-53.

[3] J. Dassow, On fully initial context-free languages, *Papers on Automata and Languages*, 10 (1988), 3-6.

[4] Al. Mateescu, Gh. Paun, Further remarks on fully initial grammars, *Acta Cybernetica*, 9, 2 (1989), 143-156.

[5] H. A. Maurer, A. Salomaa, D. Wood, A super normal form theorem for context-free grammars, *Report F63*, Institut für Inform., Techn. Univ. Graz, 1981.

[6] Gh. Paun, A note on fully initial context-free languages, *Papers on Automata and languages*, 10 (1988), 7-11.

[7] A. Salomaa, *Formal languages*, Academic Press, New York, London, 1973.

[8] S. Vicolov, A note on fully initial grammars, *Acta Cybernetica*, 10, 1-2 (1991), 113-118.

(Received November 13, 1990)

Acta Cybernetica, Tom. 10, Fasc. 1-2, Szeged, 1991

Modelling of heterogeneous multiprocessor systems with randomly changing parameters

J. Sztrik*†

Abstract

A queueing theoretic approach is developed to analyse the performance of heterogeneous multiprocessor computer systems evolving in random environments. The time intervals from the completion of the previous bus usage to the generation of a new request as well as the holding times of the common bus are assumed to be exponentially distributed random variables with parameter depending on the state of the corresponding random environment. Each processor is characterised by its own acces and service rate. The bus arbiter selects the processor to use the common bus according to a First-Come, First-Served (FCFS) discipline. Supposing that the acces rates of the processors are much greater than the corresponding service rates ("fast" arrival), it is shown that the busy period length of the bus converges weakly, under appropriate norming, to an exponentially distributed random variable. As a consequence the main steady-state performance measures, such as utilizations, throughput, mean delay time, expected waiting time, the average number of requests served during a busy period, and mean number of active processors can be calculated. Moreover, exact and approximate validation results are presented to illustrate the credibility of the proposed method.

Keywords: queueing, multiprocessor system, performance measures, weak convergence, random environments, utilization.

1 Introduction

In multiprocessor systems the contention for a common bus is one of the major factors affecting the computer performance. Several papers have been devoted to the analysis of such systems under different conditions on acces rates, the distribution function of holding times, and bus arbitration protocols (c.f., Ajmone Marsan *et al.* (1986), Bodnar and Liu (1989), Gelenbe (1989), Noyami and Sumita (1989)). More recently Ishigaki *et al.* (1990) suggested a queueing theoretic approach to analyse the system and a numerical technique was used for the evaluation of the basic

*Department of Mathematics University of Debrecen Debrecen, P.O. Box 12, Hungary, 4010

†Research done while J. Sztrik was visiting the Department of Mathematics, University of Bradford, England. The work is partially supported by István Széchenyi Foundation, Hungary, Hungarian National Science Foundation under grant OTKA-1648.

performance measures. In this work an asymptotic queueing theoretic approach is proposed to study the performance of a First-Come, First-Served (FCFS) heterogeneous single bus multiprocessor system evolving in random environments. All random times in the system are considered to be exponentially distributed, while each processor is characterised by its own acces and service rates depending on the state of the corresponding random environment. Under a heavy traffic assumption (i.e., "fast" arrivals), it is shown that the busy period length of the bus converges weakly, under appropriate norming, to an exponentially distributed rendom variable. This result facilitates the calculation of several steady-state performance measures of interest.

Note that the asymptotic technique has a widespread applicability in the field of reliability theory (c.f., Anisimov *et al* (1987), Anisimov and Sztrik (1989), Gertsbakh (1984, 1989)). Refinements in the model are often needed when the the system environment is subject to randomly occuring fluctuations which appear as changes in the parameters of the model. These fluctuations may be due to changes in the physical environment, personnel changes, alteration of computer system usage intensity, etc., (c.f., Baccelli and Makowski (1986), Gaver *et al.* (1984), Gelenbe and Rosenberg, Neuts (1978), Rosenberg *et al.* (1990), Sengupta (1990)).

2 Preliminary results

This section presents a brief survey of results (c.f., Anisimov *et al.* (1987)) to be applied in the next section.

Let $(X_\varepsilon(k), k \geq 0)$ be a Markov chain with state space

$$\bigcup_{q=0}^{m+1} X_q, \quad X_i \cap X_j = 0, \; i \neq j,$$

with $m+2$ levels of states, $i, j = 0, 1, \ldots, m+1$, defined by the transition matrix $(p_\varepsilon(i^{(q)}, j^{(z)}))$, $i^{(q)} \in X_q$, $j^{(z)} \in X_z, q, z = 0, 1, \ldots, m+1$ satisfying the following conditions:

1. $p_\varepsilon(i^{(0)}, j^{(0)}) \longrightarrow p_0(i^{(0)}, j^{(0)})$, as $\varepsilon \longrightarrow 0$, $i^{(0)}, j^{(0)} \in X_0$, and matrix $P_0 = (p_0(i^{(0)}, j^{(0)}))$ is irreducible;
2. $p_\varepsilon(i^{(q)}, j^{(q+1)}) = \varepsilon\alpha^{(q)}, j^{(q+1)}) + o(\varepsilon)$, $i^{(q)} \in X_q$, $j^{(q+1)} \in X_{q+1}$, where $\alpha^{(q)}(i^{(q)}, j^{(q+1)})$ is an appropriate transition matrix;
3. $p_\varepsilon(i^{(q)}, f^{(q)}) \longrightarrow 0$, as $\varepsilon \longrightarrow 0$, $i^{(q)}, f^{(q)} \in X_q, q \geq 1$;
4. $p_\varepsilon(i^(q), f^{(z)}) \equiv 0 \; i^{(q)} \in X_q$, $f^{(z)} \in X_z$, $\quad z - q \geq 2$.

In the sequeel the set of states X_q is called the q-th level of the chain, $q = 0, \ldots, m+1$. Let us single out the subset of states

$$\langle \alpha_m \rangle = \bigcup_{q=0}^{m} X_q.$$

Denote by $\{\pi_\varepsilon(i^{(q)}), \; i^{(q)} \in X_q\}$, $\quad q = 1, \ldots, m$ the stationary distribution of a chain with transition matrix

$$\left(\frac{p_\varepsilon(i^{(q)}, j^{(z)})}{1 - \sum\limits_{k^{(m+1)} \in X_{m+1}} p_\varepsilon(i^{(q)}, k^{(m+1)})}\right), i^{(q)} \in X_q, j^{(z)} \in X_z, \; q, z \leq m.$$

Furthermore denote by $g_\varepsilon(\langle\alpha_m\rangle)$ the steady state probability of exit from $\langle\alpha_m\rangle$, that is

$$g_\varepsilon(\langle\alpha_m\rangle) = \sum_{i^{(m)}\in X_m} \pi_\varepsilon(i^{(m)}) \sum_{j^{(m+1)}\in X_{m+1}} p_\varepsilon(i^{(m)}, j^{(m+1)}).$$

Denote by $\{\pi_o(i^{(0)}),\quad i^{(0)} \in X_0\}$ the stationary distribution corresponding to P_o and let

$$\overline{\pi}_o = \{\pi_o(i^{(0)}),\ i^{(0)} \in X_0\}, \qquad \overline{\pi}_\varepsilon^{(q)} = \{\pi_\varepsilon(i^{(q)}), i^{(q)} \in X_q\},$$

be row vectors. Finally, let the matrix

$$A^{(q)} = (\alpha^{(q)}(i^{(q)}, j^{(q+1)})),\quad i^{(q)} \in X_q,\ j^{(q+1)} \in X_{q+1},\ q = 0, \dots, m$$

defined by condition 2.

Conditions (1)-(4) enable us to compute the main terms of the asymptotic expression for $\overline{\pi}_\varepsilon^{(q)}$ and $g_\varepsilon(\langle\alpha_m\rangle)$. Namely, we obtain

$$\overline{\pi}_\varepsilon^{(q)} = \varepsilon^q \overline{\pi}_o A^{(0)} A^{(1)} \dots A^{(q-1)} + o(\varepsilon^q), \quad q = 1, \dots, m,$$

$$g_\varepsilon(\langle\alpha_m\rangle) = \varepsilon^{m+1} \overline{\pi}_o A^{(0)} A^{(1)} \dots A^{(m)} \underline{1} + o(\varepsilon^{m+1}), \tag{1}$$

where $\underline{1} = (1, \dots, 1)^*$ is a column vector, (c.f., Anisimov *et al.* (1987), pp. 141-153).

Let $(\eta_\varepsilon(t),\quad t \geq 0)$ be a Semi Markov Process (SMP) given by the embedded Markov chain $(X_\varepsilon(k),\quad k \geq 0)$ satisfying conditions (1)-(4). Let the times $\tau_\varepsilon(j^{(s)}, k^{(z)})$ - transition times from state $j^{(s)}$ to state $k^{(z)}$ - fulfill the condition

$$E \exp\{i\Theta\beta_\varepsilon\tau_\varepsilon(j^{(s)}, k^{(z)})\} = 1 + a_{jk}(s, z, \Theta)\varepsilon^{m+1} + o(\varepsilon^{m+1}), \quad (i^2 = -1)$$

where β_ε is some normalizing factor. Denote by $\Omega_\varepsilon(m)$ the instant at which the SMP reaches the $(m+1)$-th level for the first time, exit time from $\langle\alpha_m\rangle$, provided $\eta_\varepsilon(0) \in \langle\alpha_m\rangle$. Then we have:

Theorem 1 *(c.f., Anisimov et al. (1987), pp. 153) If the above (1)-(4) conditions are satisfied then*

$$\lim_{\varepsilon\longrightarrow 0} E \exp\{i\Theta\beta_\varepsilon\Omega_\varepsilon(m)\} = (1 - A(\Theta))^{-1},$$

where

$$A(\Theta) = \frac{\sum\limits_{j^{(0)},k^{(0)}\in X_o} \pi_o(j^{(0)}) p_o(j^{(0)}, k^{(0)}) a_{jk}(0, 0, \Theta)}{\overline{\pi}_o A^{(0)} A^{(1)} \dots A^{(m)} \underline{1}}.$$

Corollary 1 *In particular, if $a_{jk}(s, z, \Theta) = i\Theta m_{jk}(s, z)$ then the limit is an exponentially distributed random variable with mean*

$$\frac{\sum\limits_{j^{(0)},k^{(0)}\in X_o} \pi_o(j^{(0)}) p_o(j^{(0)}, k^{(0)}) m_{jk}(0, 0)}{\overline{\pi}_o A^{(0)} A^{(1)} \dots A^{(m)} \underline{1}}.$$

3 The Queueing Model

Consider a multiprocessor computer system in which N different processors with a common memory are connected by a single bus. A processor that generates a request to use the bus is said to be active, otherwise it is called inactive or idle. The bus arbitration protocol (selection rule) is assumed to be FCFS, that is, the arbiter selects the next processor to use the bus amongst the active ones in order of requests' arrivals. The time intervals from the completion of the previous bus usage to the generation of a new request as well as the holding times of the common bus are exponentially distributed random variables with parameter depending on the state of the corresponding random environment. Each processor is characterised by its own acces and service rate. The processors operate in a random environment governed by an ergodic Markov chain $(\xi_1(t), \quad t \geq 0)$ with state space $(1, \ldots, r_1)$ and with transition rate matrix $a^{(1)}_{i_1 j_1}$, $i_1, j_1 = 1, \ldots, r_1$, $\quad a^{(1)}_{i_1 i_1} = \sum_{j \neq i_1} a^{(1)}_{i_1 j})$. Moreover, it is assumed that each processor can have at most one outstanding request at any time, i.e., each processor can generate a new request only after the bus usage of the previous request has been completed. Whenever the environmental process is in state i_1, let $\lambda_p(i_1, \varepsilon)$ be the access rate for processor p, $p = 1, \ldots, N$, respectively. Similarly, the shared bus is supposed to operate in a random environment governed by an ergodic Markov chain $(\xi_2(t), \quad t \geq 0)$ with state space $(1, \ldots, r_2)$ and with transition rate matrix $(a^{(2)}_{i_2 j_2}$, $i_2, j_2 = 1, \ldots, r_2$, $\quad a^{(2)}_{i_2 i_2} = \sum_{j \neq i_2} a^{(2)}_{i_2 j})$. Whenever the environmental process is in state i_2, let $\mu_p(i_2)$ be the service rate for processor p, $\quad p = 1, \ldots, N$, respectively. To this end the probability that processor p generates a request in the time interval $(t, t+h)$ is $\lambda_p(i_1, \varepsilon)h + o(h)$, where $\varepsilon > 0$, $\quad i_1 = 1, \ldots r_1$, and the probability that processor p completes the bus usage in time interval $(t, t+h)$ is $\mu_p(i_2)h + o(h)$, $i_2 = 1, \ldots, r_2$, $p = 1, \ldots, N$.

All random variables and the random environment are assumeed to be independent of each other.

Let us consider the system under the heavy traffic assumption, i.e., $\lambda_p(i_1, \varepsilon) \longrightarrow \infty$ as $\varepsilon \longrightarrow 0$. For simplicity let $\lambda_p(i_1, \varepsilon) = \lambda_p(i_1)/\varepsilon$, $\quad p = 1, \ldots, N$, $i_1 = 1, \ldots, r_1$.

Denote by $Y_\varepsilon(t)$ the number of inactive processors at time t, and let

$$\Omega_\varepsilon(m) = \inf\{t : t > 0,\ Y_\varepsilon(t) = m + 1 \ / \ Y_\varepsilon(0) \leq m\},$$

i.e., the instant at which the number of inactive processors reaches the $(m+1)$-th level for the first time, provided that at the beginning their number is not greater than m, $m = 1, \ldots, N-1$. In particular, if $m = N-1$ then the bus becomes idle since there is no active processor and, hence $\Omega_\varepsilon(N-1)$ can be referred to as the busy period length of the bus.

Denote by $\pi_o(i_1, i_2 : 0; k_1, \ldots, k_N)$ the steady- state probability that $\xi_1(t)$ is in state i_1, $\xi_2(t)$ is in state i_2, there is no idle processor and the order of requests' arrival to the bus is $(k_1, \ldots, k_N)$. Similarly, denote by $\pi_o(i_1, i_2 : 1; k_2, \ldots, k_N)$ the steady-state probability that the first random environment is in state i_1, the second one is in state i_2, processor k_1 is inactive and the other processors sent their requests in order $(k_2, \ldots, k_N)$. Clearly $(k_s, \ldots, k_N) \in V_N^{N-s+1}$, $s = 1, 2$, where V_N^{N-s+1} denotes the set of all variations of order $N - s + 1$ of integets $1, \ldots, N$. Now we have:

Theorem 2 *For the system in question under the above assumptions, independently of the initial state, the distribution of the normalized random variable*

$\varepsilon^m \Omega_\varepsilon(m)$ converges weakly to an exponentially distributed random variable with parameter

$$\wedge = \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \sum_{(k_1,\ldots,k_N)\in V_N^N} \pi_o(i_1, i_2 : 1; k_2, \ldots, k_N)$$

$$\times \frac{\mu_{k_2}(i_2)}{\lambda_{k_1}(i_1)} \frac{\mu_{k_3}(i_2)}{\lambda_{k_1}(i_1) + \lambda_{k_2}(i_1)} \times \ldots \times \frac{\mu_{k_{m+1}}(i_2)}{\lambda_{k_1}(i_1) + \ldots + \lambda_{k_m}(i_1)} \frac{1}{D},$$

where

$$D = \sum_{\substack{i_1,j_1=1 \\ j_1\neq i_1}}^{r_1} \sum_{\substack{i_2,j_2=1 \\ j_2\neq i_2}}^{r_2} \sum_{(k_1,\ldots,k_N)\in V_N^N} \pi_o(i_1, i_2, 0; k_1, \ldots, k_N)$$

$$\times \frac{a^{(1)}_{i_1 j_1} + a^{(2)}_{i_2 j_2}}{(a^{(1)}_{i_1 i_1} + a^{(2)}_{i_2 i_2} + \mu_{k_1}(i_2))^2}.$$

Proof. Let us introduce the following stochastic process

$$Z_\varepsilon(t) = (\xi_1(t), \xi_2(t) : Y_\varepsilon(t); \beta_1(t), \ldots, \beta_{N-Y_\varepsilon(t)}(t))$$

where $\beta_1(t), \ldots, \beta_{N-Y_\varepsilon(t)}(t)$ denotes the indices of the active processors in the order of their request arrival to the bus. It is easy to see that $(Z_\varepsilon(t), t \geq 0)$ is a multidimensional Markov chain with state space

$$E = ((i_1, i_2 : s; k_1, \ldots, k_{N-s}), \quad i_1 = 1, \ldots, r_1, \quad i_2 = 1, \ldots, r_2,$$

$$(k_1, \ldots, k_{N-s}) \in V_N^{N-s}, s = 0, \ldots, N)$$

where $k_o = \{0\}$ by definition.
Furthermore, let

$$\langle \alpha_m \rangle = ((i_1, i_2 : s; k_1, \ldots, k_{N-s}), \quad i_1 = 1, \ldots, r_1, \quad i_2 = 1, \ldots, r_2,$$

$$(k_1, \ldots, k_{N-s}) \in V_N^{N-s}, s = 0, \ldots, m).$$

Hence our aim is to determine the distribution of the first exit time of $Z_\varepsilon(t)$ from $\langle \alpha_m \rangle$, provided that $Z_\varepsilon(o) \in \langle \alpha_m \rangle$.

It can easily be verified that the transition probabilities for the embedded Markov chain are

$$p_\varepsilon[(i_1, i_2 : s; k_1, \ldots, k_{N-s}), (j_1, i_2 : s; k_1, \ldots, k_{N-s})]$$

$$= \frac{a^{(1)}_{i_1 j_1}}{a^{(1)}_{i_1 i_1} + a^{(2)}_{i_2 i_2} + \sum_{p\neq k_1,\ldots,k_{N-s}} \lambda_p(i_1)/\varepsilon + \mu_{k_1}(i_2)}, \quad s = 0, \ldots, N-1,$$

$$p_\varepsilon[i_1,i_2:N;0),(j_1,i_2:N;0) = \frac{a^{(1)}_{i_1j_1}}{a^{(1)}_{i_1i_1}+a^{(2)}_{i_2i_2}+\sum_{p=1}^{N}\lambda_p(i_1)/\varepsilon}, \quad s=N,$$

$$p_\varepsilon[(i_1,i_2:s;k_1,\ldots,k_{N-s}),(i_1,j_2:s;k_1,\ldots,k_{N-s})]$$
$$= \frac{a^{(2)}_{i_2j_2}}{a^{(1)}_{i_1i_1}+a^{(2)}_{i_2i_2}+\sum_{p\neq k_1,\ldots,k_{N-s}}\lambda_p(i_1)/\varepsilon+\mu_{k_1}(i_2)}, \quad s=0,\ldots,N-1,$$

$$p_\varepsilon[(i_1,i_2:N;0),= \frac{a^{(2)}_{i_2j_2}}{a^{(1)}_{i_1i_1}+a^{(2)}_{i_2i_2}+\sum_{p=1}^{N}\lambda_p(i_1)/\varepsilon}, \quad s=N,$$

$$p_\varepsilon[(i_1,i_2:s;k_1,\ldots,k_{N-s}),(i_1,i_2:s+1;k_2,\ldots,k_{N-s})]$$
$$= \frac{\mu_{k_1}(i_2)}{a^{(1)}_{i_1i_1}+a^{(2)}_{i_2i_2}+\sum_{p\neq k_1,\ldots,k_{N-s}}\lambda_p(i_1)/\varepsilon+\mu_{k_1}(i_2)}, \quad s=0,\ldots,N-1,$$

$$p_\varepsilon[(i_1,i_2:s;k_1,\ldots,k_{N-s}),(i_1,i_2:s-1;;k_1,\ldots,k_{N-s+1})]$$
$$= \frac{\lambda_{k_{N-s+1}}(i_1)/\varepsilon}{a^{(1)}_{i_1i_1}+a^{(2)}_{i_2i_2}+\sum_{p\neq k_1,\ldots,k_{N-s}}\lambda_p(i_1)/\varepsilon+\mu_{k_1}(i_2)}, \quad s=1,\ldots,N-1,$$

$$p_\varepsilon[(i_1,i_2:N;0),(i_1,i_2:N-1;k)] = \frac{\lambda_k(i_1)}{a^{(1)}_{i_1i_1}+a^{(2)}_{i_2i_2}+\sum_{p=1}^{N}\lambda_p(i_1)/\varepsilon}, \quad s=N.$$

As $\varepsilon \longrightarrow 0$ this implies

$$p_\varepsilon[(i_1,i_2:0;k_1,\ldots,k_N),(j_1,i_2:0;k_1,\ldots,k_N)] = \frac{a^{(1)}_{i_1j_1}}{a^{(1)}_{i_1i_1}+a^{(2)}_{i_2i_2}+\mu_{k_1}(i_2)}, \quad s=0,$$

$$p_\varepsilon[(i_1,i_2:0;k_1,\ldots,k_N),(i_1,j_2:0;k_1,\ldots,k_N)] = \frac{a^{(2)}_{i_2j_2}}{a^{(1)}_{i_1i_1}+a^{(2)}_{i_2i_2}+\mu_{k_1}(i_2)}, \quad s=0,$$

$$p_\varepsilon[(i_1,i_2:s;k_1,\ldots,k_{N-s}),(j_1,i_2:s;k_1,\ldots,k_{N-s})] = o(1), \quad s=1,\ldots,N,$$
$$p_\varepsilon[(i_1,i_2:s;k_1,\ldots,k_{N-s}),(i_1,j_2:s;k_1,\ldots,k_{N-s})] = o(1), \quad s=1,\ldots,N,$$

$$p_\varepsilon[(i_1,i_2:0;k_1,\ldots,k_N),(i_1,i_2:1;k_2,\ldots,k_N)] = \frac{\mu_{k_1}(i_2)}{a^{(1)}_{i_1i_1}+a^{(2)}_{i_2i_2}+\mu_{k_1}(i_2)}, \quad s=0,$$

$$p_\varepsilon[(i_1,i_2:s;k_1,\ldots,k_{N-s}),(i_1,i_2:s+1;;k_2,\ldots,k_{N-s})]$$
$$= \frac{\mu_{k_1}(i_2)\varepsilon}{\sum_{p\neq k_1,\ldots,k_{N-s}}\lambda_p(i_1)}(1+o(1)), \quad s=1,\ldots,N-1.$$

This agrees with the conditions (1)-(4), but here the zero level is the set

$$((i_1,i_2:0,k_1,\ldots,k_N),(i_1,i_2:1;k_1,\ldots,k_{N-1},\ i_1=1,\ldots,r_1,\ i_2=1,\ldots,r_2,$$
$$(k_1,\ldots,k_{N-s})\in V_N^{N-s},\ s=0,1),$$

while the q-th level is the set

$$((i_1, i_2 : q+1; k_1, \ldots, k_{N-q-1}), \quad i_1 = 1, \ldots, r_1, \quad i_2 = 1, \ldots, r_2,$$
$$(k_1, \ldots, k_{N-q-1}) \in V_N^{N-q-1}).$$

Since the level 0 in the limit forms an essential class, the probabilities

$\pi_o(i_1, i_2 : 0 k_1, \ldots, k_N)$, $\pi_o(i_1, i_2 : 1; k_1, \ldots, k_{N-1})$, $\quad i_1 = 1, \ldots, r_1$, $i_2 = 1, \ldots, r_2$,

$(k_1, \ldots, k_{N-s}) \in V_N^{N-s}$, $\quad s = 0, 1$, satisfy the following system of equations

$$\begin{aligned}
\pi_o(j_1, j_2 : 0; k_1, \ldots, k_N) &= \sum_{i_1 \neq j_1} \pi_o(i_1, j_2 : 0; k_1, \ldots, k_N) a^{(1)}_{i_1 j_1} / [a^{(1)}_{i_1 i_1} + a^{(2)}_{j_2 j_2} + \mu_{k_1}(j_2)] \\
&+ \sum_{i_2 \neq j_2} \pi_o(j_1, i_2 : 0; k_1, \ldots, k_N) a^{(2)}_{i_2 j_2} / [a^{(2)}_{j_1 j_1} + a^{(2)}_{i_2 i_2} + \mu_{k_1}(i_2)] \\
&+ \pi_o(j_1, j_2 : 1; k_1, \ldots, k_{N-1}),
\end{aligned} \tag{2}$$

$$\begin{aligned}
&\pi_o(j_1, j_2 : 1; k_1, \ldots, k_{N-1}) \\
&\quad = \pi_o(j_1, j_2 : 0; k_N, k_1, \ldots, k_{+N-1}) \mu_{k_N}(j_2) / [a^{(1)}_{j_1 j_1} + a^{(2)}_{j_2 j_2} + \mu_{k_N}(j_2)].
\end{aligned} \tag{3}$$

To apply the asymptotic expressions (1), it is necessary to solve system (2), (3), subject to normalizing condition

$$\sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \sum_{(k_1, \ldots, k_N)} \{\pi_o(i_1, i_2 : 0; k_1, \ldots, k_N) + \pi_o(i_1, i_2 : 1; k_1, \ldots, k_{N-1})\} = 1.$$

Suppose this solution is known. Then by substituting it into (1) it follows that

$$\begin{aligned}
g(\langle \alpha_m \rangle) = \varepsilon^m \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \sum_{(k_1, \ldots, k_N) \in V_N^N} \pi_o(i_1, i_2 : 1; k_2, \ldots, k_N) \frac{\mu_{k_2}(i_2)}{\lambda_{k_1}(i_1)} \frac{\mu_{k_3}(i_2)}{\lambda_{k_1}(i_1) + \lambda_{k_2}(i_1)} \\
\times \ldots \times \frac{\mu_{k_{m+1}}(i_2)}{\lambda_{k_1}(i_1) + \ldots + \lambda_{k_m}(i_1)} (1 + o(1)).
\end{aligned} \tag{4}$$

Taking into account the exponentiality of $\tau_\varepsilon(j_1, j_2 : s; k_1, \ldots, k_{N-s})$ for fixed Θ it is implied that

$$E \exp\{i \varepsilon^m \Theta \tau_\varepsilon(j_1, j_2 : 0; k_1, \ldots, k_N)\} = 1 + \varepsilon^m \frac{i\Theta}{a^{(1)}_{j_1 j_1} + a^{(2)}_{j_2 j_2} + \mu_{k_1}(j_2)} (1 + o(1)),$$

$$E \exp\{i \varepsilon^m \Theta \tau_\varepsilon(j_1, j_2 : s; k_1, \ldots, k_{N-s})\} = 1 + o(\varepsilon^m), \quad s > 0.$$

Notice that $\beta_\varepsilon = \varepsilon^m$ and therefore from Corollary 1 our statement immediately follows.

However, if $\mu_p(i_2) = \mu(i_2)$, $p = 1, \ldots, N$, $i_2 = 1, \ldots, r_2$, then by substituting (3) into (2) then we get

$$\begin{aligned}\pi_o(j_1, j_2 : 0; k_1, \ldots, k_N) &= \sum_{i_1 \neq j_1} \pi_o(i_1, j_2 : 0; k_1, \ldots, k_N) a^{(1)}_{i_1 j_1} / [a^{(1)}_{i_1 i_1} + a^{(2)}_{j_2 j_2} + \mu(j_2)] \\ &+ \sum_{i_2 \neq j_2} \pi_o(j_1, i_2 : 0; k_1, \ldots, k_N) a^{(2)}_{i_2 j_2} / [a^{(1)}_{j_1 j_1} + a^{(2)}_{i_2 j_2} + \mu(i_2)] \\ + \pi(j_1, j_2 : 0; k_N, k_1, \ldots, k_{N-1}) \mu(j_2) / [a^{(1)}_{j_1 j_1} &+ a^{(2)}_{j_2 j_2} + \mu(j_2)]. \end{aligned} \tag{5}$$

Since the steady-state distributions of the governing Markov chains satisfy

$$\pi^{(1)}_{j_1} a^{(1)}_{j_1 j_1} = \sum_{i_1 \neq j_1} \pi^{(1)}_{i_1} a^{(1)}_{i_1 j_1}, \qquad \pi^{(2)}_{j_2} a^{(2)}_{j_2 j_2} = \sum_{i_2 \neq j_2} \pi^{(2)}_{i_2} a^{(2)}_{i_2 j_2}, \tag{6}$$

it can easily be verified, that the solution of (5) together with (6) is

$$\pi_o(i_1, i_2 : 0; k_1, \ldots, k_N) = B \pi^{(1)}_{i_1} \pi^{(2)}_{i_2} (a^{(1)}_{i_1 i_1} + a^{(2)}_{i_2 i_2} + \mu(i_2)),$$

$$\pi_o(i_1, i_2 : 1; k_1, \ldots, k_{N-1}) = B \pi^{(1)}_{i_i} \pi^{(2)}_{i_2} \mu(i_2),$$

where B is the normalising constant, i.e.

$$1/B = N! \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \pi^{(1)}_{i_1} \pi^{(2)}_{i_2} (a^{(1)}_{i_1 i_1} + a^{(2)}_{i_2 i_2} + 2\mu(i_2)).$$

Thus, from Th. 2 follows that $\varepsilon^m \Omega_\varepsilon(m)$ converges weakly to an exponentially distributed random variable with parameter

$$\wedge = \frac{\mu(i_2)^{m+1}}{N!} \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \sum_{(k_1, \ldots, k_N) \in V^N_N} \pi^{(1)}_{i_1} \pi^{(2)}_{i_2} \frac{1}{\lambda_{k_1}(i_1)} \frac{1}{\lambda_{k_1}(i_1) + \lambda_{k_2}(i_1)}$$

$$\times \ldots \times \frac{1}{\lambda_{k_1}(i_1) + \ldots + \lambda_{k_m}(i_1)}.$$

Consequently, the distribution of the time while the number of idle processors reaches the $(m+1)$-th level for the first time is approximated by

$$P(\Omega_\varepsilon(m) > t) = P(\varepsilon^m \Omega_\varepsilon(m) > \varepsilon^m t) \approx \exp(-\varepsilon^m \wedge t).$$

In particular, when $m = N - 1$, we get that the busy period length of the bus is asymptotically an exponentially distributed random variable with parameter

$$\varepsilon^{N-1} \wedge = \varepsilon^{N-1} \frac{\mu(i_2)^N}{N!} \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \sum_{(k_1, \ldots, k_N) \in V^N_N} \pi^{(1)}_{i_1} \pi^{(2)}_{i_2} \frac{1}{\lambda_{k_1}(i_1)} \frac{1}{\lambda_{k_1}(i_1) + \lambda_{k_2}(i_1)}$$

$$\times \ldots \times \frac{1}{\lambda_{k_1}(i_1) + \ldots + \lambda_{k_N}(i_1)}. \tag{7}$$

In the case when there are no random environments, i.e., $\mu(i_2) = \mu$, and $\lambda_p(i_1) = \lambda_p$, $i_1 = 1, \ldots, r_1$, $i_2 = 1, \ldots, r_2$, $p = 1, \ldots, N$, from (7) it follows that

$$\varepsilon^{N-1}\Lambda = \frac{\mu^N}{N!} \sum_{(k_1,\ldots,k_N)\in V_N^N} \frac{1}{\lambda_{k_1}/\varepsilon} \frac{1}{\lambda_{k_1}/\varepsilon + \lambda_{k_2}/\varepsilon} \times \ldots \times \frac{1}{\lambda_{k_1}/\varepsilon + \ldots + \lambda_{k_{N-1}}/\varepsilon}. \tag{8}$$

Finally, for the special case of totally homogeneous processors (i.e., $\lambda_p = \lambda$, $p = 1, \ldots, N$) expression (8) reduces to

$$\varepsilon^{N-1}\Lambda = \frac{1}{(N-1)!} \frac{\mu^N}{(\lambda/\varepsilon)^{N-1}}. \tag{9}$$

4 Performance Measures

This section deals with the derivation of the main steady-state performance measures relating to the heterogeneous multiprocessor model treated in the previous section.

4.1 Utilizations

The utilization U of the bus is defined as the fraction of time during which it is busy. The idle period of the bus starts when each processor is idle at the end of a service completion, and terminates when a processor generates a request. It is clear that the mean idle period length is

$$\sum_{i_1=1}^{r_1} \pi_{i_1}^{(1)} \frac{1}{\sum_{p=1}^{N} \lambda_p(i_1)/\varepsilon}.$$

Hence for U the following expression is obtained

$$U = \frac{\frac{1}{\varepsilon^{N-1}\Lambda}}{\frac{1}{\varepsilon^{N-1}\Lambda} + \sum_{i_1=1}^{r_1} \pi_{i_1} \frac{1}{\sum_{p=1}^{N} \lambda_p(i_1)/\varepsilon}}. \tag{10}$$

The bus utilization U_p of processor p is defined as the fraction of time that processor p uses the bus. Since the processors have identically distributed holding times we get

$$U_p = U \sum_{i_1=1}^{r_1} \pi_{i_1}^{(1)} \left(\lambda_p(i_1) \Big/ \sum_{k=1}^{N} \lambda_k(i_1) \right). \tag{11}$$

4.2 Throughput

The throughput γ_p of processor p is defined as the mean number of requests of processor p served per unit time. It is well-known that

$$U_p = \gamma_p b_p$$

where b_p is the mean bus usage (service) time of a request by processor p.

In this case

$$U_p = \gamma_p \sum_{i_2=1}^{r_2} \pi_{i_2}^{(2)} \frac{1}{\mu(i_2)}$$

and thus

$$\gamma_p = U_p \Bigg/ \sum_{i=1}^{r_2} \pi_{i_2}^{(2)} \frac{1}{\mu(i_2)}.$$

4.3 Mean delay and waiting times

The mean delay T_p of processor p is the average time from the instant at which a request is generated at processor p to the instant at which the bus usage of that request has been completed. In other words, T_p is the mean duration of an active state at processor p. Since the state of processor p alternates betweenn the active state of average duration T_p and the inactive state of mean duration

$$\sum_{i_1=1}^{r_1} \pi_{i_1}^{(1)} \frac{1}{\lambda_p(i_1)/\varepsilon}$$

the following relationship clearly holds

$$\gamma_p = \frac{1}{T_p + \sum_{i_1=1}^{r_1} \pi_{i_1}^{(1)} \frac{1}{\lambda_p(i_1)/\varepsilon}}.$$

Thus,

$$T_p = \frac{1}{\gamma_p} - \sum_{i_1=1}^{r_1} \pi_{i_1}^{(1)} \frac{1}{\lambda_p(i_1)/\varepsilon}.$$

Furthermore, for the mean waiting time W_p of processor p it follows that

$$W_p = T_p - \sum_{i_2=1}^{r_2} \pi_{i_2}^{(2)} \frac{1}{\mu(i_2)}.$$

4.4 Average number of requests served during a busy period

A pair of an idle period followed by an busy period is called a cycle, whose mean length is dennoted by C. Clearly,

$$C = \frac{1}{\varepsilon^{N-1}\Lambda} + \sum_{i_1=1}^{r_1} \pi_{i_1}^{(1)} \frac{1}{\sum\limits_{p=1}^{N} \lambda_p(i_1)/\varepsilon}.$$

Denote by N_p the mean number of requests of processor p served during a cycle. The throughput γ_p of processor p is then given by $\gamma_p = N_p/C$, which yields that the total number of requests served during an busy peirod is

$$\sum_{p=1}^{N} N_p = \sum_{p=1}^{N} \gamma_p C.$$

4.5 Mean number of active processors

Let us denote by $Q^{(p)}$ the steady-state probability that processor p is idle. Clearly, we have

$$Q^{(p)} = \gamma_p \sum_{i_1=1}^{r_1} \pi_{i_1}^{(1)} \frac{1}{\lambda_p(i_1)/\varepsilon}.$$

Hence, the mean number of active processors is

$$\sum_{p=1}^{N} (1 - Q^{p)}) = N - \sum_{p=1}^{N} Q^{(p)}.$$

5 Numerical Results

This section presents a number of validation experiments (c.f., Tables 1-8) examining the credibility of the proposed approximation against exact results for the performance measure of processor utilization at equilibrium. Note that an exact formula for the utilization is known only when the system is not effected by random environment and it is given (via Palm-formula) by

$$U_p^* = \frac{1}{N} \frac{\sum\limits_{k=1}^{N} \binom{N}{k} k! \rho^k}{1 + \sum\limits_{k=1}^{N} \binom{N}{k} k! \rho^k},$$

where $\rho = \frac{\lambda/\varepsilon}{\mu}$.

In this case relations (9-11)) reduce to the following approximation

$$U_p = \frac{1}{N} \frac{N!}{N! + \left(\frac{\mu}{\lambda/\epsilon}\right)^N}.$$

The following results are derived:

Table 1

$N = 3$

ρ	U_p^*	U_p
1	0.3125	0.285714286
2	0.329113924	0.326530612
2^2	0.332657201	0.332467532
2^3	0.333237575	0.333224862
2^4	0.333320592	0.333319771
2^5	0.333333169	0.333331638
2^6	0.333333125	0.333333121
2^7	0.333333307	0.333333307
2^8	0.333333333	0.333333333

Table 2

$N = 4$

ρ	U_p^*	U_p
1	0.246153846	0.24
2	0.249605055	0.249350649
2^2	0.249968310	0.249959317
2^3	0.249997756	0.249997457
2^4	0.249999999	0.249999999
2^5	0.25	0.25

Table 3

$N = 5$

ρ	U_p^*	U_p
1	0.199386503	0.198347107
2	0.199968409	0.199947930
2^2	0.199998732	0.199998372
2^3	0.199999955	0.199999949
2^4	0.199999998	0.199999998
2^5	0.2	0.2

Table 4

$N = 6$

ρ	U_p^*	U_p
1	0.166581502	0.166435506
2	0.166664473	0.166666305
2^2	0.166666623	0.166666661
2^3	0.166666666	0.166666666

Table 5

$N = 7$

ρ	U_p^*	U_p
1	0.142846715	0.1428828804
2	0.142857009	0.142856921
2^2	0.142857142	0.142857141
2^3	0.142857143	0.142857141

Table 6

$N = 8$

ρ	U_p^*	U_p
1	0.124998860	0.1249969
2	0.124999993	0.124999988
2^2	0.125	0.125

Table 7

$N = 9$

ρ	U_p^*	U_p
1	0.111110998	0.111110805
2	0.111111111	0.111111111

Table 8

$N = 10$

ρ	U_p^*	U_p
1	0.099999999	0.99999999
2	0.1	0.1

It can be observed from Tables 1-8 that the approximate values for $\{U_p\}$ are very much comparable in accuracy to those provided by the exact results for $\{U_p^*\}$. However, the computational complexity, due to the proposed approximation, has been considerably reduced. As λ/ε becomes greater that μ, the $\{U_p\}$ approximations, as expected, approach the exact values of $\{U_p^*\}$. Clearly, the greater the number of processors the less number of steps are needed to reach the exact results.

Acknowledgement. The author is very grateful to Prof. Hideaki Takagi for providing him with the paper Ishigaki *et al.* (1990). His thanks are also due to Prof. M. Arató for his helpful remarks which greatly improved the presentation.

References

[1] Ajmone Marsan M., Balbao G., Conte G. (1986) *Performance Modeling of Multiprocessor Systems,* The MIT Press, Cambridge, Massachusetts.

[2] Anisimov V. V., Zakusilo O.K., Donchenko V.S. (1987) *Elements of queueing theory and asymptotic analysis of systems,* Visha Skola, Kiev (in Russian).

[3] Anisimov V.V., Sztrik J. (1989) Asymptotic analysis of some controlled finite-source queueing systems, *Acta Cybernetica* **9** 27-38.

[4] Baccelli F., Makowski A. (1986) Stability and bounds for single server queues in random environment, *Communications in Statistics: Stochastic Models* **2** 281-292.

[5] Bodnar B.D., Liu A.C. (1989) Modeling and Performance Analysis of Single-Bus Tightly-Coupled Multiprocessors, *IEEE Transactions on Computers* **38** 464-470.

[6] Gaver D.P., Jacobs P.A., Latouche G. (1984) Finite birth-and-death models in randomly changing environments, *Advances in Applied Probability* **16** 715-731.

[7] Gelenbe E. (1988) *Multiprocessor Performance,* John Wiley and Sons, New York.

[8] Gelenbe E. Rosenberg C. (1990) Queues with slowly varying arrival and service processes, *Management Science* **36** 928-937.

[9] Gertsbakh I.B. (1984) Asymptotic methods in reliability theory; A review, *Advances in Applied Probability* **16** 147-175.

[10] Gertsbakh I.B. (1989) *Statistical Reliability Theory,* Marcel Dekker, Inc. New York.

[11] Ishigaki M., Takagi H., Takahashi Y., Hasewaga T. (1990) Throughput and Fairness Analysis of Prioritized Multiprocessor Bus Arbitration Protocols, *Research Report* RT 0051 08/17/90, IBM Research, Tokyo Research Laboratory.

[12] Neuts M. (1978) The M/M/1 Queue with Randomly Varying Arrival and Service Rates, *Opsearch* **15** 139-157.

[13] Nogami S., Sumita S., (1989) Approximate Analysis of Memory Contetion in a Multiprocessor System with a Single Bus, *Electronics and Communications in Japan,* Part 1, Vol. 27, No. 8 65-74.

[14] Rosenberg C., Mazumdar R., Kleinrock L. (1990) On the Analysis of Exponential Queueing Systems with Randomly Changing Arrival Rates: Stability Conditions and Finite Buffer Scheme with Resume Level, *Performance Evaluation* **11** 283-292.

[15] Sengupta B. (1990) A queue with service interruptions in an alternating random environment, *Operations Research* **38** 308-318.

(Received April 4, 1991)

Acta Cybernetica, Tom. 10, Fasc. 1-2, Szeged, 1991

Modelling of a Communication System Evolving in a Random Environment

by J. Sztrik * L. Lukashuk ‡

Abstract

This paper is concerned with a queueing model to analyse the asymptotic behaviour of a finite-source communication system with a receiver containing multiple processors of the same kind. The source and processing times of each message are supposed to be exponentially distributed random variables with parameter depending on the state of a varying environment. Assuming that the arrivals of the messages are "fast" compared to their service, it is shown that the time to the first system failure converges in distribution, under appropriate norming, to an exponentially distributed random variable.

Keywords: queueing, communication system, reliability, weak convergence.

1 Introduction

Performance evaluation of information system development has become more complex as the size and complexity of the system has increased, see Takagi (1990). Reliability is certainly the most important characteristic for communication networks. The measure of greatest interest is the distribution of the time to the first system failure. It is well-known that the majority of the problems can be treated by the help of Semi-Markov Processes (SMP) or Semi-Regenerative Processes (SRP). Since the failure-free operation time of the system correponds to sojourn time problems we can use the results obtained for SMP, cf. Ushakov (1985), Osaki et al. (1987). If the exit from a given subset of the state space is a "rare" event, that is, it occurs with a small probability it is natural to investigate the asymptotic behaviour of the sojourn time in that subspace, see Gertsbakh (1984, 1989), Keilson (1979), Rukhin and Hsieh (1987).

This paper is concerned with a queueing model to analyse the asymptotic behaviour of a finite-source communication system with a receiver containing multiple processors of the same kind. The source and processing times of each message

*Department of Mathematics, University of Debrecen, Hungary

†Department of Applied Statistics, Kiev State University, USSR

‡Research done while J. Sztrik was visiting the Department of Mathematics, University of Bradford, England and supported by István Széchenyi Foundation, Hungary, Universitas-90 Foundation, University of Debrecen, Hungary; furthermore while L. Lukashuk was visiting the Department of Mathematics, University of Debrecen, Hungary and supported by Universitas-90 Foundation.

are supposed to be exponentially distributed random variables with parameter depending on the state of a varying environment. Assuming that the arrivals of the messages are "fast" compared to their service, it is shown that the time to the first system failure converges in distribution, under appropriate norming, to an exponentially distributed random variable.

2 Preliminary results

In this section a brief survey is given of the most related theoretical results, mainly due to Anisimov, to be applied later on.

Let $(X_\varepsilon(k), k \geq 0)$ be a Markov chain with state space

$$\bigcup_{q=0}^{m+1} X_q, \quad X_i \cap X_j = 0, \quad i \neq j,$$

defined by the transition matrix $\|p_\varepsilon(i^{(q)}, j^{(z)})\|$ satisfying the following conditions:

1. $p_\varepsilon(i^{(0)}, j^{(0)}) \longrightarrow p_0(i^{(0)}, j^{(0)}), as \quad \varepsilon \longrightarrow 0,$

   $i^{(0)}, j^{(0)} \in X_0$, and $P_0 = \|p_0(i^{(0)}, j^{(0)})\|$ is irreducible;

2. $p_\varepsilon(i^{(q)}, j^{(q+1)}) = \varepsilon \alpha^{(q)}(i^{(q)}, j^{(q+1)}) + o(\varepsilon), \quad i^{(q)} \in X_q, \quad j^{(q+1)} \in X_{q+1};$

3. $p_\varepsilon(i^{(q)}, f^{(q)}) \longrightarrow 0$, as $\varepsilon \longrightarrow 0, \quad i^{(q)}, f^{(q)} \in X_q, q \geq 1;$

4. $p_\varepsilon(i^{(q)}, f^{(z)}), \equiv 0, \quad i^{(q)} \in X_q, f^{(z)} \in X_z, \quad z - q \geq 2.$

In the sequel the set of states X_q is called the q-th level of the chain, $q = 1, \ldots, m+1$. Let us single out the subset of states

$$\langle \alpha_m \rangle = \bigcup_{q=0}^{m} X_q.$$

Denote by $\{\pi_\varepsilon(i^{(q)}), i^{(q)} \in X_q\}, q = 1, \ldots, m$ the stationary distribution of a chain with transition matrix

$$\left\| \frac{p_\varepsilon(i^{(q)}, j^{(z)})}{1 - \sum\limits_{k^{(m+1)} \in X_{m+1}} p_\varepsilon(i^{(q)}, k^{(m+1)})} \right\|, i^{(q)} \in X_q, j^{(z)} \in X_z, q, z \leq m,$$

furthermore denote by $g_\varepsilon(\langle \alpha_m \rangle)$ the steady state probability of exit from $\langle \alpha_m \rangle$, that is

$$g_\varepsilon(\langle \alpha_m \rangle) = \sum_{i^{(m)} \in X_m} \pi_\varepsilon(i^{(m)}) \sum_{j^{(m+1)} \in X_{m+1}} p_\varepsilon(i^{(m)}, j^{(m+1)}).$$

Denote by $\{\pi_0(i^{(0)}), i^{(0)} \in X_0\}$ the stationary distribution corresponding to P_0 and let

$$\bar{\pi}_0 = \{\pi_0(i^{(0)}), i^{(0)} \in X_0\}, \quad \bar{\pi}_\varepsilon^{(q)} = \{\pi_\varepsilon(i^{(q)}), i^{(q)} \in X_q\}$$

be row vectors. Finally, let

$$A^{(q)} = \|\alpha^{(q)}(i^{(q)}, j^{(q+1)})\|, \quad i^{(q)} \in X_q, j^{(q+1)} \in \lambda_{q+1}, q = 0, \dots, m$$

defined by Condition 2.

Conditions (1)-(4) enables us to compute the main terms of the asymptotic expression for $\pi_\varepsilon^{(q)}$ and $g_\varepsilon(\langle\alpha_m\rangle)$. Namely, we obtain

$$\overline{\pi}_\varepsilon^{(q)} = \varepsilon^q \overline{\pi}_0 A^{(0)} A^{(1)} \dots A^{(q-1)} + o(\varepsilon^q) \quad q = 1, \dots, m,$$

$$g_\varepsilon(\langle\alpha_m\rangle) = \varepsilon^{m+1} \overline{\pi}_0 A^{(0)} A^{(1)} \dots A^{(m)} \underline{1} + o(\varepsilon^{m+1}), \tag{1}$$

where $\underline{1} = (1, \dots, 1)^*$ is a column vector, see Anisimov et al. (1987) pp. 141-153. Let $(\eta_\varepsilon(t), \quad t \geq 0)$ be a SMP given by the embedded Markov chain $(X_\varepsilon(k) \quad k \geq 0)$ satisfying conditions (1)-(4). Let the times $\tau_\varepsilon(j^{(s)}, k^{(z)})-$ transition times from state $j^{(s)}$ to state $k^{(z)}-$ fulfil the condition

$$\mathbf{E} \exp\{i\Theta\beta_\varepsilon\tau_\varepsilon(j^{(s)}, k^{(z)})\} = 1 + a_{jk}(s, z, \Theta)\varepsilon^{m+1} + o(\varepsilon^{m+1}), \quad (i^2 = -1)$$

where β_ε is some normalizing factor. Denote by $\Omega_\varepsilon(m)$ the instant at which the SMP reaches the $m+1$-th level for the first time, exit time from $\langle\alpha_m\rangle$, provided $\eta_\varepsilon(0) \in \langle\alpha_m\rangle$. Then we have:

Theorem 1 *(cf. Anisimov et al. (1987) pp. 153) If the above conditions are satisfied then*

$$\lim_{\varepsilon \to 0} \mathbf{E} \exp\{i\Theta\beta_\varepsilon\Omega_\varepsilon(m)\} = (1 - A(\Theta))^{-1},$$

where

$$A(\Theta) = \frac{\sum_{j^{(0)}, k^{(0)} \in X_0} \pi_0(j^{(0)}) p_0(j^{(0)}, k^{(0)}) a_{jk}(0, 0, \Theta)}{\overline{\pi}_0 A^{(0)} A^{(1)} \dots A^{(m)} \underline{1}}.$$

Corollary 1 *In particular, if $a_{jk}(s, z, \Theta) = i\Theta m_{jk}(s, z)$ then the limit is an exponentially distributed random variable with mean*

$$\frac{\sum_{j^{(0)}, k^{(0)} \in X_0} \pi_0(j^{(0)}) p_0(j^{(0)}, k^{(0)}) m_{jk}(0, 0)}{\overline{\pi}_0 A^{(0)} A^{(1)} \dots A^{(m)} \underline{1}}.$$

3 The mathematical model

Let us consider a communication system consisting of N sources of information and n processors of the same kind at the receiver. The whole system is assumed to operate in a random environment governed by an ergodic Markov chain $(\xi(t), t \geq 0)$ with state space $(1, \dots, r)$ and with transition density matrix $(a_{ij}, i, j = 1, \dots, r, a_{ii} = \sum_{j \neq i} a_{ij})$. Whenever the environmental process is in state i the probability that an active source generates a message in the time interval $(t, t+h)$ is $\lambda(i, \varepsilon)h + o(h)$. Each message is transmitted to a receiver where the service immediately starts if there is an idle processor, otherwise a queueing line is formed. The service discipline is First Come-First Served (FCFS). Whenever the environmental process is in state i the probability that the processing of a given message is completed in time interval $(t, t+h)$ is $\mu(i)h + o(h)$. If a given source has sent a message it stays idle and it cannot generate other one. After being serviced each message immediately returns to its source which hence becomes active. All random variables involved her and the random environment are supposed to be independent of each other.

In practical applications it is very important to know the distribution of time until the receiver becomes empty.

Let us consider the system unnder the assumption of "fast" arrivals, i.e., $\lambda(i, \varepsilon) \longrightarrow \infty$ as $\varepsilon \longrightarrow 0$. For simplicity let $\lambda(i, \varepsilon) = \lambda(i)/\varepsilon$. Denote by $Y_\varepsilon(t)$ the number of active sources at time t, and let

$$\Omega_\varepsilon(m) = \inf\{t : t > 0, Y_\varepsilon(t) = m + 1 | Y_\varepsilon(0) \le m\},$$

that is, the instant at which the number of active sources reaches the $m+1$-th level for the first time, provided that at the begining their number is not greater than m, $m = 0, \ldots, N-1$. In the following $\Omega_\varepsilon(m)$ is referred to as the time to the first system failure. In particular, if $m = N-1$ than the receiver becomes empty.

Denote by $(\pi_k, k = 1, \ldots, r)$ the steady-state distribution of the governing Markov chain $(\xi(t), t \ge 0)$. Now we have:

Theorem 2 *For the system in question under the above assumptions, independently of the initial state, the distribution of the normalized random variable $\varepsilon^m \Omega_\varepsilon(m)$ converges weakly to an exponentially distributed random variable with parameter*

$$\Lambda = \frac{n}{m!} \prod_{s=1}^{m} \min(n, N-s) \sum_{i=1}^{r} \pi_i \frac{\mu(i)^{m+1}}{\lambda(i)^m}.$$

Proof. It is easy to see that the process

$$Z_\varepsilon(t) = (\xi(t), Y_\varepsilon(t))$$

is a two-dimensional Markov chain with state space

$$E = ((i, x), \quad i = 1, \ldots, r, \quad s = 0, \ldots N).$$

Furthermore, let

$$\langle \alpha_m \rangle = ((i, s), \quad i = 1, \ldots, r, \quad s = 0, \ldots, m).$$

Hence our aim is to determine the distribution of the first exit time of $Z_\varepsilon(t)$ from $\langle \alpha_m \rangle$, provided that $Z_\varepsilon(\circ) \in \langle \alpha_m \rangle$.

It can easily be verified that the transition probabilities in any time interval $(t, t+h)$ are the following:

$$(i, s) \xrightarrow{h} \begin{cases} (j, s) & a_{ij}h + o(h), & i \ne j, \\ (i, s+1) & \min(n, N-s)\mu(i)h + o(h) & s = 0, \ldots, N-1, \\ (i, s-1) & (s\lambda(i)/\varepsilon)h + o(h), & s = 1, \ldots, N. \end{cases}$$

In addition, the sojourn time $\tau_\varepsilon(i, s)$ of $Z_\varepsilon(t)$ in state (i, s) is exponentially distributed with parameter $a_{ii} + s\lambda(i)/\varepsilon + \min(N, n-s)\mu(i)$. Thus, the transition probabilities for the embedded Markov chain are

$$p_\varepsilon[(i,s),(j,s)] = \frac{a_{ij}}{a_{ii} + s\lambda(i)/\varepsilon + \min(n, N-s)\mu(i)}, \quad s = 0,\dots,N,$$

$$p_\varepsilon[(i,s),(i,s+1)] = \frac{\min(n, N-s)\mu(i)}{a_{ii} + s\lambda(i)/\varepsilon + \min(n, N-s)\mu(i)}, \quad s = 0,\dots,N-1,$$

$$p_\varepsilon[(i,s),(i,s-1)] = \frac{s\lambda(i)/\varepsilon}{a_{ii} + s\lambda(i)/\varepsilon + \min(n, N-s)\mu(i)}, \quad s = 1,\dots,N.$$

As $\varepsilon \longrightarrow 0$ this implies

$p_\varepsilon[i,0),(j,0)] = \frac{a_{ij}}{a_{ii}+n\mu(i)}, \quad s = 0,$

$p_\varepsilon[(i,s),(j,s)] = o(1), \quad s = 1,\dots,N,$

$p_\varepsilon[(i,0),(i,1)] = \frac{n\mu(i)}{a_{ii}+n\mu(i)}, \quad s = 0,$

$p_\varepsilon[(i,s),(i,s+1)] = \frac{\min(n,N-s)\mu(i)\varepsilon}{s\lambda(i)}(1 + o(\varepsilon)), \quad s = 1,\dots,N-1,$

$p_\varepsilon[(i,s),(i,s-1)] \longrightarrow 1, \quad s = 1,\dots N.$

This agrees with the conditions (1)-(4), but here the zero level is the set $((i,0),(i,1), i = 1,\dots,r)$ while the q-th level is $((i,q+1), i = 1,\dots,r)$. Since the level 0 in the limit forms an essential class, the probabilities $\pi_0(i,0), \pi_0(i,1)$, $i = 1,\dots,r$ satisfy the following system of equations

$$\pi_0(j,0) = \sum_{i \neq j} \pi_0(i,0) a_{ij}/(a_{ii} + n\mu(i)) + \pi_0(j,1) \tag{2}$$

$$\pi_0(j,1) = \pi_0(j,0) n\mu(j)/(a_{jj} + n\mu(j)). \tag{3}$$

By substituting (3) to (2) we get

$$\pi_0(j,0) a_{jj}/(a_{jj} + n\mu(j)) = \sum_{i \neq j} \pi_0(i,0) a_{ij} \setminus (a_{ii} + n\mu(i)). \tag{4}$$

Since

$$\pi_j a_{jj} = \sum_{i \neq j} \pi_i a_{ij},$$

from (3) and (4) we have

$$\pi_0(i,0) = B\pi_i(a_{ii} + n\mu(i)), \quad \pi_0(i,1) = B\pi_i n\mu(i),$$

where B is the normalizing constant, i.e.

$$1/B = \sum_{i=1}^{r} \pi_i [a_{ii} + 2n\mu(i)].$$

By using (1) it is easy to show that the probability of exit from $\langle \alpha_m \rangle$ is

$$\begin{aligned} g_\varepsilon(\langle \alpha_m \rangle) &= \varepsilon^m n B \sum_{i=1}^{r} \pi_i \mu(i) \prod_{s=1}^{m} \frac{\min(n, N-s)\mu(i)}{s\lambda(i)} (1 + o(1)) \\ &= \frac{\varepsilon^m n B}{m!} \prod_{s=1}^{m} \min(n, N-s) \sum_{i=1}^{r} \pi_i \frac{\mu(i)^{m+1}}{\lambda(i)^m} (1 + o(1)). \end{aligned}$$

Taking into account the exponentiality of $\tau_\varepsilon(j, s)$ for fixed Θ we have

$$\mathbf{E} \exp \{i\varepsilon^m \Theta \tau_\varepsilon(j, 0)\} = 1 + \varepsilon^m \frac{i\Theta}{a_{jj} + n\mu(j)} (1 + o(1))$$

$$\mathbf{E} \exp \{i\varepsilon^m \Theta \tau_\varepsilon(j, s)\} = 1 + o(\varepsilon^m), \quad s > 0.$$

Notice that $\beta_\varepsilon = \varepsilon^m$ and therefore from Corollary 1 we immediately get the statement that $\varepsilon^m \Omega_\varepsilon(m)$ converges weakly to an exponentially distributed random variable with parameter

$$\Lambda = \frac{n}{m!} \prod_{s=1}^{m} \min(n, N-s) \sum_{i=1}^{r} \pi_i \frac{\mu(i)^{m+1}}{\lambda(i)^m},$$

which completes the proof.

Consequently, the distribution of the time to the first system failure can be approximated by

$$P(\Omega_\varepsilon(m) > t) = P(\varepsilon^m \Omega_\varepsilon(m) > \varepsilon^m t) \approx \exp(-\varepsilon^m \Lambda t),$$

i.e. $\Omega_\varepsilon(m)$ is asymptotically an exponenetially distributed random variable with parameter $\varepsilon^m \Lambda$. In particular, for $m = N - 1$ we have

$$\varepsilon^{N-1} \Lambda = \frac{\varepsilon^{N-1} n!}{(N-1)!} n^{N-n} \sum_{i=1}^{r} \pi_i \frac{\mu(i)^N}{\lambda(i)^{N-1}} = \frac{n!}{(N-1)!} n^{N-n} \sum_{i=1}^{r} \pi_i \frac{\mu(i)^N}{(\lambda(i)/\varepsilon)^{N-1}}.$$

In the case when there is no random environment we get

$$\varepsilon^{N-1} \Lambda = \frac{n!}{(N-1)!} n^{N-n} \frac{\mu^N}{(\lambda/\varepsilon)^{N-1}}.$$

References

[1] Anisimov V. V., Zakusilo O.K., Donchenko V.S. (1987): Elements of queueing theory and asymptotic analysis of systems, Visha Skola, Kiev (in Russian).

[2] Gertsbakh I.B. (1984) Asymptotic methods in reliability theory; A review, Adv. Appl. Math. 16 157-175.

[3] Gertsbakh I.B. (1989) Statistical Reliability Theory, Marcel Dekker, New York.

[4] Keilson J. (1979) Markov chain models - Rarity and exponentiality, Springer-Verlag, Berlin.

[5] Osaki S., Cao J. (1987) (Editors) Reliability Theory and Applications, World Scientific, Singapore.

[6] Rukhin A.L., Hsieh H.K. (1987) Survey of Soviet Works on Reliability Theory, Statistical Science 2 484-503.

[7] Ushakov A. I. (1985) (Editor) Reliability of Complex Systems, Radio i Svjaz, Moscow (in Russian).

[8] Takagi H. (1990) Bibliography on Performance Evaluation, Stochastic Analysis of Computer and Communication Systems, H. Takagi (Editor), IFIP 1990, North-Holland, Amsterdam.

(Received December 11, 1990)

Acta Cybernetica, Tom. 10, Fasc. 1-2, Szeged, 1991

Weak dependecnies in the relational datamodel

V. D. Thi * N. K. Anh †

1 Introduction

One of the main concepts in relational database theory is the full family of functional dependencies, that was first axiomatized by W. W. Armstrong [1]. The full family of dual, strong and weak dependencies have also been introduced and axiomatized in [2,3,4]. The logical structures of them have also been investigated in [5,6,7,8,9,10].

In this paper, we give some results, that are related to weak dependencies. We give a necessary and sufficient condition for $W_R^+ = Y$, and then we construct an effective combinatorial algorithm to detemine irredundant relation R' for an arbitrary given relation R such that $R' \subseteq R, W_{R'}^+ = W_R^+$. Connections between dependencies are investigated also.

2 Definitions and axioms

Definition 2.1 *Let Ω be a finite set of attributes, and $R = \{h_1, \ldots, h_m\}$ be a relation over $\Omega, A, B \subseteq \Omega$. Then we say that B weakly depends A in R (denote $A \xrightarrow[R]{w} B$) if*

$$(\forall h_i, h_j \in R) \quad ((\forall a \in A) \quad (h_i(a) = h_j(a)) \longrightarrow (\exists b \in B) \quad h_i(b) = h_j(b)))$$

B functionally depends A in R (denote $A \xrightarrow[R]{f} B$) if

$$(\forall h_i, h_j \in R) \quad ((\forall a \in A) \quad (h_i(a) = h_j(a)) \longrightarrow (\forall b \in B) \quad (h_i(b) = h_j(b)))$$

B dually depends A in R (denote $A \xrightarrow[R]{d} B$) if

$$(\forall h_i, h_j \in R) \quad ((\exists a \in A) \quad (h_i(a) = h_j(a)) \longrightarrow (\exists b \in B) \quad h_i(b) = h_j(b))).$$

Let $W_R^+ = \{(A, B) : A, B \neq \emptyset$ and $A \xrightarrow[R]{w} B\}$ and $\overline{X} = \Omega \setminus X$ for any $X \subset P(\Omega)$.

$$F_R = \{(A, B) : A \xrightarrow[R]{f} B\} \text{ and } D_R = \{A, B) : A \xrightarrow[R]{d} B\}.$$

*This research was partially suported by Hungarian National Foundation of Research Grant no. 2575.

†Centre for System and Management Research. National Centre for Scientific Research of Vietnam. Lieu giai. Ba dinh. Hanoi. Vietnam.

Definition 2.2 *Let Ω be a finite set, and denote $P(\Omega)$ its power set, $P^+(\Omega) = P(\Omega) \setminus \{\emptyset\}$. Let $Y \subseteq P^+(\Omega) \times P^+(\Omega)$. Then we say that Y satisfies the w^+-axioms, if for all $A, B, C, D \in P^+(\Omega)$*

(w_1^+) $(A,B) \in Y,\ A \subseteq C, B \subseteq D \longrightarrow (C,D) \in Y$;

(w_2^+) $A, B \in P^+(\Omega), ((\forall X \in P^+(\Omega)) \quad (A \subseteq X \subseteq \overline{B} \longrightarrow (X, \overline{X}) \in Y)) \longrightarrow (A,B) \in Y$.

Let $Y \subseteq P(\Omega) \times P(\Omega)$. We say that Y satisfies the A-axiom if for all $A \subseteq \Omega$, there is an $E(A)$ such that

(f_1) $A \subseteq E(A)$, and $\forall B \subseteq E(A) \longrightarrow (A,B) \in Y$;
(f_2) $(C,D) \in Y, C \subseteq E(A) \longrightarrow D \subseteq E(A)$.

Y satisfies the B-axiom if for all $B \subseteq \Omega$, there is an $E(B)$ such that

(d_1) $B \subseteq E(B)$, and $\forall A \subseteq E(B) \longrightarrow (A,B) \in Y$;
(d_2) $(C,D) \in Y, C \not\subseteq E(B) \longrightarrow D \not\subseteq E(B)$.

Definition 2.3 *Let $Y \subseteq P^+(\Omega) \times P^+(\Omega)$. We say that Y is an w^+-family over Ω if Y satisfies the w^+-axioms.*

Let $Y \subseteq P(\Omega) \times P(\Omega)$. We say that Y is an $f - (d-)$family over Ω if Y satisfies the $A - (B-)$axiom.

Theorem 2.4 *[3]. Let $Y \subseteq P^+(\Omega) \times P^+(\Omega)$. If R is a relation over Ω, then W_R^+ satisfies the w^+-axioms. Conversely, if Y satisfies the w^+-axioms, then there is a relation R over Ω such that $Y = W_R^+$.*

3 The family of weak dependencies.

Definition 3.1 *Let Y be an w^+-family, and R be a relation over Ω. Then we say that R represents Y iff $W_R^+ = Y$.*

Definition 3.2 *Let Y be an w^+-family, over Ω, and $X \in P^+(\Omega)$. We say that $(X, \overline{X})$ is an Ω-dependency of Y if $(X, \overline{X}) \in Y$.*

Denote by $M(Y)$ the set of all Ω-dependencies of Y. We say that X is an Ω-left side of Y if $(X, \overline{X}) \in M(Y)$, and X is an Ω-right side of Y if $(\overline{X}, X) \in M(Y)$. Denote $GF(Y)$ the set of all Ω-left sides of Y, and $GD(Y)$ the set of all Ω-right sides of Y. It is obvious that $GF(Y)$ and $GD(Y)$ does not contain $\emptyset, \Omega$.

Theorem 3.3 *Let $G \subseteq P^+(\Omega) \setminus \{\Omega\}$. There exist exactly one w^+-family Y so that $GF(Y) = G$, where*

$$Y = \{(A,B) \in P^+(\Omega) \times P^+(\Omega) : (\forall X \in P^+(\Omega)) \quad (A \subseteq X \subseteq \overline{B} \longrightarrow X \in G)\}$$

Proof. In order to prove the theorem, we need the following lemma.

Lemma 3.4 *Let Y be an w^+-family over Ω. Then $(A,B) \in Y$ iff $(\forall X \in P^+(\Omega)) \quad (A \subseteq X \subseteq \overline{B} \longrightarrow (X, \overline{X}) \in M(Y))$.*

Proof. If $(A,B) \in P^+(\Omega) \times P^+(\Omega)$ satisfies

$$(\forall X \in P^+(\Omega)) \quad (A \subseteq X \subseteq \overline{B} \longrightarrow (X, \overline{X}) \in M(Y)),$$

then $(A,B) \in Y$ by (w_2^+). Conversely, if $(A,B) \in Y$, then

$$(\forall X \in P^+(\Omega)) \quad (A \subseteq X \subseteq \overline{B} \longrightarrow A \subseteq X, B \subseteq \overline{X} \longrightarrow (X, \overline{X} \in M(Y) \text{ by } (w_1^+)).$$

The lemma is proved.

□

We have to show that Y is an w^+-family. By the definition of Y, it is obvious that Y satisfies (w_2^+), where $M(Y) = \{(X,\overline{X}) \in Y\} = \{X,\overline{X}) : X \in G\}$, and $GF(Y) = G$. We have to prove that Y satisfies (w_1^+). For all $A,B,C,D \in P^+(\Omega), (A,B) \in Y, A \subseteq C, B \subseteq D, (\forall X \in P^+(\Omega)) \quad (C \subseteq X \subseteq \overline{D} \longrightarrow A \subseteq C \subseteq X \subseteq \overline{D} \subseteq \overline{B} \longrightarrow X \in G$ by $(A,B) \in Y)) \longrightarrow (C,D) \in Y$.

Now, we suppose that there is an w^+-family Y' so that $GF(Y') = G$, then $M(Y') = \{(X,\overline{X}) : X \in G\} = M(Y)$. Hence $Y' = Y$ by lemma 3.4. The proof is complete.

□

Corollary 3.5 *Let $G \subseteq P^+(\Omega) \setminus \{\Omega\}$. There exist exactly one w^+-family Y so that $GD(Y) = G$, where*

$$Y = \{(A,B) \in P^+(\Omega) \times P^+(\Omega) : (\forall X \in P^+(\Omega)) \quad (A \subseteq X \subseteq \overline{B} \longrightarrow \overline{X} \in G)\}$$

Definition 3.6 *[4]. Let $R = \{h_1, \ldots, h_m\}$ be a relation over Ω. Let*

$$E_{i,j} = \{a \in \Omega : h_i(a) = h_j(a), 1 \leq i < j \leq m\}.$$

We call $E_{i,j}$ the equality set of R. Denote by E_R the family of all equality sets of R. Practically, it is possible that $\emptyset \in E_R$, and there are some $E_{i,j}$, which are equal to each other. According to the definition of relation, we have $\Omega \notin E_R$. Let

$$\begin{aligned} M_R &= \{E_{i,j} \neq \emptyset : \text{ if } E_{p,q}, E_{s,t} \in M_R, \text{ then } E_{p,q} \neq E_{s,t}\} \\ &= \{A_1, \ldots, A_k : A_i \neq A_j \text{ for } i \neq j \text{ and } A_i \neq \emptyset \text{ for } i = \overline{1,k}\}. \end{aligned}$$

It is obvious that k is the number of elements of M_R, and all elements of M_R are not equal to each other. It is obvious that $A_i \notin \{\emptyset, \Omega\}$ for $i = \overline{1,k}$.

Theorem 3.7 *Let Y be a w^+-family, and R be a relation over Ω. Then R represents Y if and only if $GF(Y) = P^+(\Omega) \setminus (M_R \cup \{\Omega\})$.*

Proof. By theorem 3.3, it is easy to see that R represents Y iff $GF(W_R^+) = GF(Y)$. Consequently, we only must prove that $GF(W_R^+) = P^+(\Omega) \setminus (M_R \cup \{\Omega\})$.

It is obvious that $GF(W_R^+)$ does not contain $\emptyset$ and Ω. If $X \in P^+(\Omega) \setminus (M_R \cup \{\Omega\})$, then $X \notin \{\emptyset, \Omega)$ and $X \neq E_{i,j}$ for $1 \leq i < j \leq m$. We have $(\forall h_i, h_j \in R) \quad ((\forall a \in X) \quad (h_i(a) = h_j(a)) \longrightarrow X \subset E_{i,j}, X \neq E_{i,j}$ and $E_{i,j} \neq \emptyset$ by $X \neq \emptyset \longrightarrow (\exists b \in \overline{X}) \quad (h_i(b) = h_j(b)))$.

Hence $(X,\overline{X}) \in W_R^+$ holds and we obtain $P^+(\Omega) \setminus (M_R \cup \{\Omega\}) \subseteq GF(W_R^+)$. Conversely, if $X \in GF(W_R^+)$, then $X \notin \{\emptyset, \Omega\}$ and $(X,\overline{X}) \in W_R^+$.

If $(h_i, h_j \in R)$ $((\exists a \in X)$ $(h_i(a) \neq h_j(a)))$, then $X \neq E_{i,j} \neq \emptyset$. If $(h_i, h_j \in R)$ $((\forall a \in X)$ $(h_i(a) = h_j(a)) \longrightarrow (\exists b \in \overline{X})$ $(h_i(b) = h_j(b)))$, then $X \neq E_{i,j}$. Hence $X \neq E_{i,j}$ holds for $1 \leq i < j \leq m$, and we obtain

$$GF(W_R^+) \subseteq P^+(\Omega) \setminus (M_R \cup \{\Omega\}).$$

The theorem is proved.

□

Definition 3.8 *[10]. Let $R = \{h_1, \ldots, h_m\}$ be a relation over Ω. Let*

$$N_{i,j} = \{a \in \Omega : h_i(a) \neq h_j(a), 1 \leq i < j \leq m\}.$$

We call $N_{i,j}$ the non-equality set of R. Denote by N_R the family of all non-equality sets of R. Practically, it is possible that $\Omega \in N_R$, and there are some $N_{i,j}$, which are equal to each other. According to the definition of relation, we have $\emptyset \notin N_R$. Let

$$\begin{aligned} S_R &= \{N_{i,j} : \text{ if } N_{p,q}, N_{s,t} \in S_R, \text{ then } N_{p,q} \neq N_{s,t}\} \\ &= \{B_1, \ldots, B_k : B_i \neq B_j \text{ for } i \neq j\}. \end{aligned}$$

It is obvious that k is the number of elements of S_R, and all elements of S_R are not equal to each other. It is obvious that $B_i \neq \{\emptyset\}$ for $i = \overline{1,k}$.

Corollary 3.9 *Let Y be an w^+-family, and R be a relation over Ω. Then R represents Y if and only if $GD(Y) = P^+(\Omega) \setminus (S_R \cup \{\Omega\})$.*

The next proposition shows that from given any w^+-family Y, we can construct one simple non-empty relation R such that $W_R^+ = Y$.

Proposition 3.10 *Let Y be an w^+-family over $\Omega, GF(Y)$ be a set of all Ω-left sides of Y, and let $M = P^+(\Omega) \setminus (GF(Y) \cup \{\Omega\})$.*

If $|M| = 0$ then R is relation for any one-element. If $|M| \geq 1$ then we assume that $M = \{A_1, \ldots, A_k\}$, we set $R = \{h_1, h_2, \ldots, h_{2k-1}, h_{2k}\}$ as follows:

$$\text{for } i = 1, \ldots k : \forall a \in \Omega \quad h_{2i-1}(a) = 2i - 1$$

$$h_{2i}(a) = \begin{cases} 2i - 1 & \textit{if } a \in A_i \\ 2i & \textit{otherwise} \end{cases}$$

Then R represents Y.

Proof. If $|M| = 0$ then $GF(Y) = P^+(\Omega) \setminus \{\Omega\}$. So $(X, \overline{X}) \in Y$ for all $X \in P^+(\Omega) \setminus \{\Omega\}$ and we have $Y = P^+(\Omega) \times P^+(\Omega)$ by (w_2^+). Thus $W_R^+ = Y$ stands for any one-element relation and $R \neq \emptyset$. If $|M| \geq 1$ then it is obvious that $R \neq \emptyset$ holds. Clearly, $E_R = M \cup \{\emptyset\}$. Hence $M = M_R$ holds and we have $GF(Y) = P^+(\Omega) \setminus (M_R \cup \{\Omega\})$. By Theorem 3.7 we obtain $W_R^+ = Y$. The proposition is proved.

□

We say that R is w^+-irredundant relation if $R' \subset R$ imply $W_{R'}^+ \neq W_R^+$. We give an effective algorithm, which determines for a given arbitrary relation R a relation R' such that $R' \subseteq R, W_{R'}^+ = W_R^+$ and R' is irredundant.

Algorithm 3.11 *Let $R = \{h_1, \dots, h_m\}$ be a relation over Ω.*

Step 1. From given relation R we construct $E_R = \{E_{i,j}$ is an equality set of $R, 1 \le i < j \le m\}$.

Step 2. From E_R we construct $M_R = \{E_{i,j} \neq \emptyset :$ if $E_{p,q}, E_{s,t} \in M_R$, then $E_{p,q} \neq E_{s,t}\}$. Assume that $M_R = \{A_1, \dots, A_k\}$.

We construct sets of index pairs, as follows: Let

$$I_1 = \{(i,j) : E_{i,j} = A_1\}, \dots, I_k = \{(i,j) : E_{i,j} = A_k\}.$$

Denote by l_i the number of elements of I_i, where $i = 1, \dots, k$.

Denote $I_q^1(p)$ and $I_q^2(p)$ the first and second indicies of p-th pair in I_q, where $q = 1, \dots, k$ and $1 \le p \le l_q$. After that we perform the program IRREDUNDANT. Then $R' = \{h_i : i \in C\}$ is an w^+-irredundant relation such that $R' \subseteq R$, and $W^+_{R'} = W^+_R$.

Proof. It is obvious that $R' \subseteq R$. According to the construction of the algorithm, it can be seen that after we perform the program, $I_t \neq \emptyset$ holds for $t = 1, \dots, k$. On the other hand, by theorem 3.7 we have $W^+_{R'} = W^+_R$. By procedure delete (i, j), program deletes all redundant rows of R. Thus, R' is an w^+-irredundant relation. The proof is complete. □

We have $I_1[1 : l_1], \dots, I_k[1 : l_k]$.

```
Program IRREDUNDANT;
begin
C := ∅;
for q := 1 to k do
      for p := 1 to l_q do
         for s := 1 to 2 do
            if I_q^s(p) ∉ C then
               begin
                  t := q;
                  while t ≤ k do
                     begin
                        r := 1;
                        while (I_t^1(r) = I_q^s(p) or I_t^2(r) = I_q^s(p)) and r ≤ l_t do
                              r := r + 1;
                        if r = l_t + 1 then
                                    begin C := C ∪ I_q^s(p); t := k + 2
                                    end
                                    else t := t + 1;
                     end;
                  if t = k + 1 then
                     for t := q to k do
                        for r := 1 to l_t do
                           begin
                              if I_t^2(r) = I_q^s(p) then begin
                                                            delete (I_t^1(r), I_q^s(p));
                                                            l_t := l_t - 1
                                                         end;
                              if I_t^1(r) = I_q^s(p) then begin
```

```
                                        delete (I_q^s(p), I_t^2(r));
                                        l_t := l_t - 1
                                    end;
                    end;
            end;
    end;
```

Remark 3.12 *It can be seen that each Step of algorithm 3.11 requires time polynomial in the number of rows and columns of R. Consequently, the time complexity of Algorithm 3.11 is polynomial in $|R|$ and $|\Omega|$, where by $|R|$ and $|\Omega|$ the number of elements of R and Ω.*

It is easy to see that R', which is constructed in Algorithm 3.11 is an w^+-irredundant relation, and has maximal cardinality.

It can be seen that if R is any w^+-irredundant relation, and R represents Y, then $\sqrt{2|M|} < |R| < 2|M|$, where $M = P^+(\Omega) \setminus (GF(Y) \cup \{\Omega\}) \neq \emptyset$, and $|R| = 1$ when $M = \emptyset$.

4 Connections between dependencies.

Claim 4.1 *[3]. Let R be a relation over Ω and let $A, B \subseteq \Omega$. Then we have*

$A \xrightarrow[R]{f} B$ *iff* $(\forall b \in B)\ (A \xrightarrow[R]{w} \{b\})$;

$A \xrightarrow[R]{d} B$ *iff* $(\forall a \in B)\ (\{a\} \xrightarrow[R]{w} B)$.

We have obtained that W_R^+ uniquely determines F_R and D_R.

Definition 4.2 *Let F be an f-family over Ω, and $(A, B) \in F$. Then we say that (A, B) is a maximal right side dependency of F if $\forall B' : B \subseteq B', (A, B') \in F \longrightarrow B' = B$. Denote by $M(F)$ the set of maximal right side dependencies of F. We say that $B (B \subseteq \Omega)$ is a maximal right side of F iff there is an A so that $(A, B) \in M(F)$. Denote $G(F)$ the set of maximal riht sides of F. A family G of subset of Ω is closed under intersection iff $A, B \in G$ imply $A \cap B \in G$. Denote M^+ the set $\{\cap M' : M' \subseteq M\}$. We say that M generates G iff $M^+ = G$.*

Theorem 4.3 *[1]. Let F be an f-family over Ω. The $G(F)$ is closed under intersection. Conversely, if G is any family of subset of Ω, which is closed under intersection, then there exists exactly one f-family F such that $G(F) = G$, where $F = \{(A, B) : \forall C \in G : A \subseteq C \longrightarrow B \subseteq C\}$.*

Definition 4.4 *Let D be a d-family over Ω, and $(A, B) \in D$. Then we say that (A, B) is a maximal left side dependency of D if $\forall A' : A \subseteq A', (A', B) \in D \longrightarrow A' = A$. Denote by $M(D)$ the set of maximal left sides dependencies of D. We say that $A (A \subseteq \Omega)$ is a maximal left side of D iff there is an B so that $(A, B) \in M(D)$. Denote $G(D)$ the set of maximal left sides of D. A family G of subset of Ω is called d-semilattice iff G contains $\emptyset, \Omega$ and $A, B \in G$ imply $A \cap B \in G$.*

Theorem 4.5 *[2]. Let D be an d-family over Ω. Then $G(D)$ is a d-semilattice over Ω. Conversely, if G is any d-semilattice, then there exists exactly one d-family D such that $G(D) = G$, where $D = \{(A, B) : \forall C \in G : A \not\subseteq C \longrightarrow B \not\subseteq C\}$.*

Theorem 4.6 *Let Y be an w^+-family over Ω. Then $D(Y) = \{(A,B) : \forall a \in A, (\{a\}, B) \in Y\}$ is an d-family over Ω and $G(D) = (P^+(\Omega) \setminus GD(Y) \setminus \{\Omega\})^+ \cup \{\emptyset\}$. $(D = D(Y))$*

Proof. It is easy to see that $D(Y)$ satisfies B-axiom. So $D(Y)$ is an d-family over Ω. Clearly, $G(D)$ is an d-semilattice over Ω. It is obvious that $G(D)$ contains $\emptyset, \Omega$. Set $\overline{GD(Y)} = P^+(\Omega) \setminus GD(Y)$, clearly, $(\overline{GD(Y)} \setminus \{\Omega\})^+$ contains Ω (by convention $\cap\emptyset = \Omega$). Now, we assume that $X \neq \emptyset, \Omega$ and if $X \in \overline{GD(Y)}$ then $(\overline{X}, X) \notin Y$. Set $X_1 = \{a \in \Omega : (\{a\}, X) \in D\} = \{a \in \Omega : (\{a\}, X) \in Y\}$. We have $X \subseteq X_1$, if we suppose that $X \neq X_1$ and choose an element a from $(X_1 \setminus X)$ then $(\{a\}, X) \in Y$ and $a \notin X$. So $(\overline{X}, X) \in Y$ by (w_1^+), this contradicts $(\overline{X}, X) \notin Y$. Hence, $X = X_1$ and $X \in G(D)$. We obtain $(\overline{GD(Y)} \setminus \{\Omega\})^+ \cup \{\emptyset\} \subseteq G(D)$.

Conversely, if $X \in G(D)$ and $X \notin \{\emptyset, \Omega\}$, then $X = \{a \in \Omega : (\{a\}, X) \in Y\}$. If we assume that $\forall Z \in \overline{GD(Y)} : X \supset Z$ then $X = \Omega$ by (w_2^+). So this contradicts $X \neq \Omega$. Consequently, there is an $Z \in \overline{GD(Y)}$ such that $X \subseteq Z$.

If there is an $Z \in \overline{GD(Y)}$ such that $X = Z$, then $X \in (\overline{GD(Y)} \setminus \{\Omega\})^+$.

Conversely, we set $H = \{Z \in \overline{GD(Y)} : X \subset Y\} = \{Z_1, \ldots, Z_k\}$. We have $X \subseteq \bigcap_{i=1}^{k} Z_i$. Let us choose an element a from $\bigcap_{i=1}^{k} Z_i$ then $(\{a\}, X) \in Y$ by (w_2^+). So, we have $\bigcap_{i=1}^{k} Z_i \subseteq X$. Thus, $X = \bigcap_{i=1}^{k} Z_i$ and we obtain $X \in (\overline{GD(Y)} \setminus \{\Omega\})^+$.

The theorem is proved.

□

Corollary 4.7 *Let Y be an w^+-family over $\Omega, \overline{GF(Y)} = P^+(\Omega) \setminus GF(Y), C = \bigcap_{X \in \overline{GF(Y)}} X$. Then $F(Y) = \{(A,B) : \forall b \in B, (A, \{b\}) \in Y\} \cup \{(\emptyset, D) : D \subseteq C\}$ is an f-family over Ω and $G(F) = (GF(Y) \setminus \{\Omega\})^+$.*

Remark. It is easy to see that $F(Y)$ satisfies A-axiom and we have $E(\emptyset) = C$.

References

[1] W. W. Armstrong; Dependency Structures of Data Base Relationships, Information Processing 74, North-Holland Publ. Co. (1974) pp. 580-583.

[2] G. Czédli; Függőségek relációs adatbázis modellben, Alkalmazott Matematikai Lapok 6 (1980) pp. 132-143.

[3] G. Czédli; On dependencies in the Relational Model of Data, Elektronische Informationsverarbeitung und Kybernetik, EIK 17(1981) 2/3, pp. 103-112.

[4] J. Demetrovics; Relációs adatmodell logikai és strukturális vizsgálata, MTA-SZTAKI Tanulmányok, Budapest, 114 (1980) pp. 1-97.

[5] J. Demetrovics, Gy. Gyepesi; On the functional dependency and some generalizations of it, Acta Cybernetica V/3 (1988) pp. 295-305.

[6] J. Demetrovics, V. D. Thi: Some results about functional dependencies, Acta Cybernetica VIII/3 (1988) pp. 273-278.

[7] J. Demetrovics, V.D. Thi; Relations and minimal keys, Acta Cybernetica VIII/3 (1988) pp. 279-285.

[8] V. D. Thi; Minimal keys and antikeys, Acta Cybernetica Tom 7, Fasc. 4 (1986) pp. 361-371.

[9] V. D. Thi; Strong dependencies and *s*-semilattices, Acta Cybernetica VIII/2 (1987) pp. 175-202.

[10] V. D. Thi; Logical dependencies and irredundant relations, Computers and Artificial Intelligence, Vol. 7, 1988, No. 2, pp. 165-184.

(Received February 12, 1991)

Acta Cybernetica, Tom. 10, Fasc. 1-2, Szeged, 1991

Covering Morphisms and Unique Minimal D-Schemes *

R. Tindell
Stevens Institute of Technology
Hoboken, NJ 07030 USA

1 Introduction

In this paper we answer the following question: given a D-scheme F, is there a (unique) smallest D-scheme within the strong equivalence class of F? In addition to providing an affirmative answer to this question, we provide a description of a reduction process leading from a D-scheme to its minimization over the class of D-schemes.

Since we are interested in D-schemes in this paper, we restrict our discussion to what, in the terminology of Elgot [CE], could be referred to as biscalar schemes whose outdegrees are bounded by 2. The paper is organized as follows. Section 2 gives the basic definitions of digraphs, schemes, and (homo)morphisms for these two classes and introduces the class of D-schemes. Section 3 recalls the definitions necessary to state the geometric characterization of D-schemes from [BT] and introduces the notion of strong behavior and strong equivalence for schemes. Section 4 develops the basic properties of morphisms between schemes. Section 5 discusses the minimization process over the class of all schemes and states and proves the main theorem of the paper, Theorem 5.1. Section 6 is the final section and details the consequences of the main theorem.

2 Basic Definitions

A *directed graph*, or *digraph*, is a 4-tuple $H = (V, E, s, t)$ where V is a set whose elements are called the *vertices*, or *nodes*, of H; E is a set whose elements are called the *edges* of H; and s and t are functions from E to V called the *source* and *target* functions, respectively, for H. We require that V and E be disjoint. A *subdigraph* of H is a digraph $H' = (V', E', s', t')$ such that V' and E' are subsets, respectively, of V and E and s' and t' are the restrictions, respectively, of s and t. If x is an edge with $s(x) = u$ and $t(x) = v$, then we say that x is an *outedge* of u and an *inedge* of v. The *indegree* (respectively, *outdegree*) in H of a node u is the number of inedges (respectively, outedges) of u in H. A *homomorphism* from digraph $H = (V, E, s, t)$ to digraph $H' = (V', E', s', t')$ is a function h from $V \cup E$ to $V' \cup E'$ such that $h(V) \subseteq V'$, $h(E) \subseteq E'$, and for each edge x of H, $h(s(x)) = s'(h(x))$

*Dedicated to the memory of Calvin C. Elgot and $h(t(x)) = t'(h(x))$. A *digraph isomorphism* is a bijective homomorphism. It is a simple matter to prove that the inverse of a bijective homomorphism is itself a homomorphism.

Let H be a digraph. A *path* in H from node u is an alternating sequence $P = u_0x_1u_1x_2u_2\ldots$ of nodes and edges of H such that $u = u_0$ and for $i \geq 1$, $s(x_i) = u_{i-1}$ and $t(x_i) = u_i$. If P is finite, we require that the last term of the sequence be a node, say $u' = u_n$, and refer to P as a path (of length n) from u to u'; we say that u' is *accessible* in H from u if such a path exists. If $P = u_0x_1\ldots x_mu_m$ and $P' = u'_0x'_1\ldots x'_nu'_n$ are paths in H with $u_m = u'_0$, then the *composite* $P \cdot P'$ of P and P' is the path $u_0x_1\ldots x_mu_mx'_1u'_1\ldots x'_nu'_n$.

The digraph H is *strongly connected* if for any two nodes u, v of H there is a path in H from u to v. For any node u of H, the *strong component* of u is the subdigraph $F[u, v]$ of H made up of all nodes and edges lying on closed paths from u.

For the remainder of the paper we fix a pair $\Gamma = (\Omega, \Pi)$ of disjoint sets, referring to elements of Ω as *operator symbols* and to the elements of Π as *predicate symbols*. A Γ-*flowchart scheme*, or more briefly a *scheme*, is a 6-tuple $F = (V, E, s, t, \lambda, b)$ such that:

1. (V, E, s, t) is a finite digraph (also denoted F).
2. the nodes of F have outdegree at most 2.
3. F has exactly one node e of outdegree 0 (called the *exit* of F).
4. b is a node of F (called the *begin* of F).
5. λ is a function from $V \setminus \{e\} \cup E$ to $\Omega \cup \Pi \cup \{1, 2\}$, called the *labeling function*, satisfying:
   a) if $x \in E$ then $\lambda(x) \in \{1, 2\}$;
   b) if node u has unique outedge x then $\lambda(u) \in \Omega$ and $\lambda(x) = 1$.
   c) if node u has distinct outedges x and y, then $\lambda(u) \in \Pi$ and $\lambda(x) \neq \lambda(y)$.

Let $F = (V, E, s, t, \lambda, b)$ and $F' = (V', E', s', t', \lambda', b')$ be schemes and let e and e' be the exits of F and F', respectively. A *scheme morphism* from F to F' is a digraph homomorphism h from F to F' preserving begins, ends and labels — that is, such that $h(b) = b'$, $h(e) = e'$ and for each $z \in (V \cup E) \setminus \{e\}$, $\lambda'(h(z)) = \lambda(z)$. A *scheme isomorphism* is a bijective scheme homomorphism. If v is the target of an edge with source u and label i, then we shall refer to node v as the i-*successor* of node u in scheme F. We shall also use the notation b_F and e_F for the begin and exit, respectively, of a scheme F.

The simplest examples of schemes are the trivial scheme and the atomic schemes. The *trivial scheme* T consists of a single point (which is thus the begin and the exit) and no edges. For each $f \in \Omega$, the *atomic scheme* determined by f (which we also perversely denote by f) consists of a single edge whose source, labeled by f, is the begin and whose target is the exit.

We now introduce basic operations used in building schemes which are analogous to the programming operations of concatenation, if-then-else, and while-do. The *composite* $F \cdot G$ of schemes F and G is the scheme obtained from the disjoint union of F and G by identifying the exit of F with the begin of G; all labels remain the same, with the point of identification retaining the label of the begin of G if such a label exists, which is to say provided that $G \neq T$. The begin of $F \cdot G$ is the begin of F; since the exit of G is now the only point of outdegree 0, it is the exit of $F \cdot G$. It should be observed that the trivial scheme is an identity for the composition operation. For each $\pi \in \Pi$, the *alternation* by π of schemes F,G is the scheme $\pi(F,G)$ constructed as follows: form the disjoint union of F, G, and a new vertex u labeled π; add new edges x,y labeled 1 and 2, respectively, with $s(x) = s(y) = u$, $t(x)$ the begin of F, and $t(y)$ the begin of G; and then identify the exit of F with that of G. Finally, given $\pi \in \Pi$ and $i \in \{1,2\}$, the *while* by π, i of scheme F, denoted $wh(\pi, i, F)$, is constructed as follows: add to F two new points u and v and two new edges x and y with $s(x) = s(y) = u$, $t(x) = v$, and $t(y)$ the begin of F; identify the exit of F with u; label u by π, y by i, and x by $3 - i$; and designate u as the begin.

A scheme G is a *D-scheme*, or "simple while scheme", if it may be constructed from the trivial scheme and atomic schemes by a finite sequence of applications of the composition, alternation and while operations.

3 The characterization theorem for D-schemes

The definition given above for D-schemes was essentially algebraic in that it specified the generators (T and the atomic schemes) and operations (composition, alternations, and whiles) for the class of such schemes. We now develop definitions which will be used throughout the paper and state the geometric characterization of D-schemes due to the author and Bloom [BT].

The *trace* of a path

$$P = u_0 x_1 u_1 \ldots$$

in scheme F is the string

$$\lambda(u_0)\lambda(x_0)\lambda(u_1)\ldots$$

if P is infinite, and the string

$$\lambda(u_0)\lambda(x_1)\ldots\lambda(u_{n-1})\lambda(x_n)$$

if P is of length n. Notice that we did not include the label of the final vertex of a finite path, and thus the trace is well defined even if the final vertex is the (unlabeled) exit node. A *successful* path of F is a path from the begin to the exit. The *strong behavior* of a scheme F is the set of traces of both the successful paths and the infinite paths from the begin of F. Two schemes are *strongly equivalent* if they have the same strong behavior.

A scheme is *accessible* if every node is accessible from the begin, is *coaccessible* if the exit is accessible from every node, and is *biaccessible* if it is both accessible and coaccessible. We point out the obvious fact that a biaccessible scheme is one in which every node lies on a successful path. One may establish by a straightforward induction on the number of operations used in building a D-scheme that every D-scheme is biaccessible. Let A be a subdigraph of a scheme F. A path $P = u_0 x_1 \ldots x_n u_n$ is said to be *A-simple* if there is at most one value of i, $0 \leq i \leq n$, for which u_i is in A. If the A-simple path P as above has initial vertex the begin of F and final vertex in A, then P is an *entry path* to A and u_n is an *entry point* of A; if on the other hand P has initial vertex in A and final vertex the exit of F, P is an *exit path* from A and u_0 is an *exit point* of A.

A *cycle* in a scheme F is a subdigraph of F made up of the edges and nodes of some positive length *simple closed path*, by which we mean a path in which the only repetition of nodes is due to the initial and final nodes being the same. A *bipath* in F from node u to node $v \neq u$ is a subdigraph of F made up of the nodes and edges lying on some pair of simple paths in F from u to v which overlap only at u and v. A *simple path* is one in which there is no repetition of nodes.

Let F be a scheme. Then F is said to be *reducible* if every cycle in F has a unique entry point, F is *coreducible* if every cycle has a unique exit point, and is *bireducible* if each cycle contains a point which is both its unique entry point and its unique exit point. Finally, F has the *bipath exit property* if for any bipath B from u to v in F, v is the unique exit point of B.

We are now prepared to state the geometric characterisation theorem for D-schemes.

Theorem 3.1 (BT) *A scheme G is a D-scheme if and only if G is biaccessible, bireducible and satisfies the bipath exit property.*

A D-scheme may be viewed as having a natural "block structure", which we will make precise after the development of some additional definitions and intermediate results. Let F be a coaccessible scheme, u a node of F, and S a set of nodes of F. S is a *bottleneck set* rel u if S contains neither u nor the exit and every path from u to the exit contains at least one point of S. If no proper subset of S is a bottleneck set rel u we say that S is a *minimal bottleneck set* rel u. If S consists of a single point v, then v is said to be a *bottleneck* rel u. If u is the begin of F, we omit reference to u and speak of a bottleneck or bottleneck set without further qualification. If v is a bottleneck rel u in F, or v is the exit of F and $u \neq v$, the *segment of F from u to v* is the subdigraph of F made up of the points and edges lying on $\{v\}$-simple paths in F from u to v. It is clear that, except possibly for the node v, every node of $F[u,v]$ has the same outdegree in $F[u,v]$ as in F, and v has outdegree 0 in $F[u,v]$. Thus $F[u,v]$ may be viewed as a scheme by designating u as the begin and labeling the nodes as in F, with the exception that the exit v of $F[u,v]$ recieves no label. Note that a segment is always nontrivial. A segment $F[u,v]$ is a *block* of F if for any edge x of F, $t(x) \in F[u,v] - \{u,v\}$ implies $s(x) \in F[u,v] - \{v\}$; and if b_F is in $F[u,v]$ then b_F is either u or v. We will use the notation $F[u]$ for $F[u, e_F]$. Note that bottleneck sets, segments and blocks are defined only for coaccessible schemes.

The next proposition, whose simple proof we omit, deals with the "inheritance" of geometric properties by segments and blocks.

Proposition 3.1 *Let v be a bottleneck rel u in F. Then*

a) v is the unique exit of $F[u,v]$;

b) if $F[u,v]$ is a block, then u is the unique entry point of $F[u,v] - \{v\}$;

c) if F is a reducible, then $F[u,v]$ is reducible.

d) if F is a D-scheme and $F[u,v]$ is a block, then $F[u,v]$ is a D-scheme.

As noted above, every block of a D-scheme is itself a D-scheme. More can be shown. We will say that scheme G is obtained by a *block substitution* from a coaccessible scheme F if for some coaccessible scheme H and block $F[u,v]$ of F, G is isomorphic to the scheme constructed as follows: delete $F[u,v] - \{u,v\}$ from F and form the disjoint union of the result with H; identify the begin of H with u and the exit of H with v; label the edges and the nonexit nodes of H as in H; and label the nodes and edges from $(F - F[u,v]) \cup \{v\}$ as in F. Note that if H is the trivial scheme T, then u and v are identified and the resulting node retains the label of v in F. The definition of "block" ensures that the result is well-defined as a scheme; in particular, the target function on edges will not attempt to map an edge of F which was retained in the new scheme to a point of $F[u,v] - \{u,v\}$. If F and H are members of some class of coaccessible schemes, we say that G is obtained by a block substitution within the class; if the result of a block substitution within a given coaccessible class always produces another scheme within the class, we say that the class is *closed under block substitution*. Note that the class of all coaccessible schemes is closed under block substitution. The characterisation theorem may be used to show that the class of D-schemes is closed under block substitution. In fact, the class of D-schemes may be characterised as the smallest class of schemes closed under block substitution and containing T, f, $f \cdot g$, $\pi(f, g)$, $wh(\pi, 1, f)$, and $wh(\pi, 2, f)$ for all $\pi \in \Pi$ and $f, g \in \Omega$.

4 Coverings

We now turn to the basic properties of morphisms between schemes.

Proposition 4.1 *Let H be a morphism from scheme F to scheme G and let u and v be points of F and G respectively. If $h(u) = v$, then H maps the outedges of u in F bijectively onto those of v in G.*

Proof. Immediate from the following direct consequences of definitions: u and $v = h(u)$ must have the same outdegrees in their respective schemes; h preserves edge labels; the outedges of a node in a scheme are enumerated by their labels.

If h is a morphism from F to G and $P = u_0x_1u_1x_2\ldots$ is a path in F, the *image* of P under h is the path $h(P) = h(u_0)h(x_1)h(u_1)h(x_2)\ldots$ in G. We shall then refer to P as a u_0*-lift* of the path $h(P)$. The following consequence of Proposition 4.1 is the direct analogue of a basic property of covering spaces in topology.

Proposition 4.2 *If h is a morphism from F to G then for any node v of G, any path Q from v in G, and any point u of F with $h(u) = v$, Q has a unique u-lift in F.*

Proof. When Q is of finite length n, Proposition 4.1 provides the basis step ($n = 1$) for a proof by induction on n; moreover, since every path of length $n > 1$ may be written as a path of length $n - 1$ followed by a path of length 1, the inductive step is immediate. If $Q = v_0y_1\ldots$ is infinite then for each $i \geq 1$ let Q_i be the initial segment $v_0y_1\ldots y_iv_i$ of Q. If P_i is the unique u-lift of Q_i, then for any $j < i$, the initial segment of P_i of path length j is a u-lift of Q_j and hence equals P_j. Thus the sequence $P_1, P_2, \ldots$ is an infinite sequence of paths with P_{i+1} an extension of P_i for $i \geq 1$ and hence determines an infinite path P which is the unique u-lift of Q.

Proposition 4.3 *If there is a morphism from F to G then F and G have the same strong behaviors.*

Proof. Let h be a morphism from F to G. Since morphisms must preserve begins, P is a path from the begin in F if and only if $h(P)$ is a path from the begin in G. Moreover, since the exit of a scheme is its unique node of outdegree 0 and morphisms preserve outdegrees, P terminates at the exit of F if and only if $h(P)$ terminates at the exit of G. Thus P is a successful path in F if and only if its image is a successful path in G. Since P and $h(P)$ have the same trace, the proof is complete.

Proposition 4.4 *If F and G are accessible schemes, then there is at most one morphism from F to G, and any such morphism must be surjective.*

Proof. Let g and h be morphisms from F to G. For any node u of F, there exists at least one path from the begin to u and thus we may define the *distance* from the begin of F to u as the length of a shortest path in F from the begin to u. We now prove by induction on the distance from the begin to u that $g(u) = h(u)$ for all nodes u of F. The only node of distance 0 from the begin is the begin itself and by the definition of morphisms, g and h both map the begin of F onto the begin of G; so the basis step is established. For the inductive step, let $P = u_0x_1 \ldots x_nu_n$ be a shortest path from the begin of F to u. Then, since u_{n-1} has smaller distance from the begin than $u = u_n$, $g(u_{n-1}) = h(u_{n-1})$. If i is the label of x_n in F, then $g(x_n)$ and $h(x_n)$ must both be outedges of the same node in G with the same label in G and hence must be the same edge y of G. It then follows that g and h map the target u_n of edge x_n onto the target of the edge y and thus $g(u) = h(u)$, and the proof of the uniqueness of morphisms is complete.

We now prove that every morphism h from F to G is surjective. Let v be a point of G. Since G is accessible, there is a path Q in G from the begin to v and thus Q has a unique b_F-lift P. It then follows that h maps the terminal point of P onto v so that v is in the image of h. If y is an edge of G, then since the source of y in G is in the image of h, Proposition 4.1 implies that y is in the image of h, and we have shown that h is surjective.

Proposition 4.5 *Let h be a surjective morphism from F to G and let C be a cycle in G. Then there exists a cycle C' in F with $h(C') = C$.*

Proof. Let $Q = v_0y_1 \ldots y_nv_0$ be a simple closed path determining C, let u_0 be a point of F with $h(u_0) = v_0$, and let m be the number of nodes in F. If Q' is the m-fold composition of Q with itself, then the u_0-lift P' of Q' cannot be a simple path since its length exceeds the number of nodes in F. We may then choose a contiguous subsequence P of P' such that P is a simple closed path in F. It then follows that $h(P)$ is a closed path all of whose nodes and lines appear in Q. If u is a node occuring in P, it has a unique outedge x in P and $h(x)$ is an outedge of $v = h(u)$ in G. Since $h(x)$ is an edge appearing in Q and u has a unique outedge in Q, $h(x)$ is the outedge of u in Q. It then follows that if u is in $h(P)$ the unique successor of u in Q is in $h(P)$ and thus that the subgraph of F determined by $h(P)$ is the same as that determined by Q, and the proof is complete.

We now turn to the question of which of the geometric properties of accessible schemes defined in section 3 are preserved by morphisms. It will be convenient to refer to a surjective morphism h from F to G as a *covering* of G by F and hence to say that F *covers* G and that F is a *cover* of G. In the situations where the domain scheme F is a D-scheme, we shall refer to h as a *D-covering* and to F as a *D-cover* of G.

We next note that neither the bipath exit property nor reducibility is preserved by coverings. Let F be the scheme having vertices b_F, v_1, v_2, e_F and satisfying

1. $\lambda(b_F) = \nu \in \Pi, \lambda(v_1) = \pi \in \Pi$, and $\lambda(v_2) = f \in \Omega$;
2. v_i is the i-successor of b_F for $i = 1, 2$;
3. v_1 has 1-successor e_F and 2-successor v_2;
4. e_F is the 1-successor of v_2.

Then F is covered by the D-scheme $\nu(\pi(T, f), f)$ and F does not satisfy the bipath exit property. If G is the scheme obtained from F by changing the 1-successor of v_2 from e_F to v_1, then G is not reducible and is covered by the D-scheme $\nu(T, f) \cdot wh(\pi, 2, f)$. The property of coreducibility is however preserved by coverings.

Proposition 4.6 *If F is coreducible and h is a covering of G by F, then G is coreducible.*

Proof. Let C be a cycle in G. Then by 4.5 there is a cycle C' in F with $h(C') = C$. If P is an exit path from C with initial point u, then for any u' in C' with $h(u') = u$, the u'-lift of P is an exit path from C'. Since C' has a unique exit point in F, it follows that C has a unique exit point in G.

One obvious consequence of proposition 4.6 is that coreducibility is a necessary condition for a biaccessible scheme to be covered by a D-scheme; it is also sufficient. Proposition 4.6 and its converse for biaccessible schemes is a generalisation of Kasai's theorem which uses coreducibility to characterise the schemes with the same strong behavior as a D-scheme [K]. We will not give a separate proof of the converse of 4.6 since it is a corollary of the main theorem of this paper, Theorem 5.1 of the next section. We now establish an important lemma and state without proof some straightforward results detailing relationships between morphisms and the D-operations.

Lemma 4.1 *Let h be a covering of G by a coaccessible scheme F and let v be a bottleneck rel u in F. Then $h(v)$ is a bottleneck rel $h(u)$ in G and $h(F[u,v]) = G[h(u), h(v)]$.*

Proof. To show that $h(v)$ is a bottleneck rel $h(u)$ in G, let Q be a path from $h(u)$ to the exit of G and consider the u-lift Q' of Q. Since Q' is a path from u to the exit of F, v must lie on Q' and therefore $h(v)$ lies on $h(Q') = Q$.

We now turn to showing that $h(F[u,v]) = G[h(u), h(v)]$. Since the u-lift of any $\{h(v)\}$-simple path in G from $h(u)$ is a $\{v\}$-simple path in F, $G[h(u), h(v)]$ is a subdigraph of $h(F[u,v])$. To prove that $h(F[u,v])$ is equal to $G[h(u), h(v)]$, we need only show that the image under h of any $\{v\}$-simple path in F from u to v is $\{h(v)\}$-simple. Suppose to the contrary that $P = v_0x_1 \dots x_nv_n$ is $\{v\}$-simple, $v_0 = u$, $v_n = v$, and $h(v_i) = h(v)$ for some $i < n$. If Q is a simple path from $h(v)$ to e_G, Q' is the v_i-lift of Q, and $P' = v_0x_1 \dots x_iv_i$, then $P' \cdot Q'$ is a path from u to the exit of F which does not contain v. This contradiction completes the proof.

Proposition 4.7 *Let π be a predicate symbol and j an element of $\{1,2\}$. If F_i covers G_i, $i = 1,2$, then*

a) $wh(\pi, j, F_1)$ covers $wh(\pi, j, G_1)$;

b) $\pi(F_1, F_2)$ covers $\pi(G_1, G_2)$; and

c) $F_1 \cdot F_2$ covers $G_1 \cdot G_2$.

Proposition 4.8 *If $F = wh(\pi, j, F')$ covers G then there is a scheme G' such that F' covers G' and $G = wh(\pi, j, G')$.*

Proposition 4.9 *If F_1 and F_2 are nontrivial coaccessible schemes such that $F_1 \cdot F_2$ covers G, then there is a bottleneck v of G such that F_1 covers $G[b_G, v]$ and F_2 covers $G[v]$.*

Proposition 4.10 *Let v be a bottleneck of coaccessible scheme G. If F_1 covers $G[b_G, v]$ and F_2 covers $G[v]$, then $F_1 \cdot F_2$ covers G.*

5 Minimal Schemes

In this section we consider the question of the existence of unique minimal schemes within strong equivalence classes. If one is minimizing over the class of all coaccessible schemes, then such schemes exist since a scheme F may be viewed as a finite automaton over the alphabet $\Omega \times \{1\} \cup \Pi \times \{1,2\}$ as follows: let b_F be the start and e_F be the unique final state, replace the label i of an edge x by the ordered pair (α, i), where α is the label of the source of x, and remove the labels of the nodes. Since the infinite strings in the strong behavior of a coaccessible scheme are determined by the finite strings, it is clear that strong equivalence is then the same as automaton equivalence. Thus the construction of the minimum scheme $F\mathbb{M}$ strongly equivalent to F is essentially the same as for finite automata and has as vertices the behavior-equivalence classes of nodes of F. Moreover, the function sending a node to its equivalence class induces a morphism of F onto $F\mathbb{M}$. As in the case of finite automata, two schemes are strongly equivalent if and only if they have isomorphic minimal schemes.

As noted in the previous section, there are D-schemes which cover schemes which are not D-schemes. In fact, the two examples given there were "minimization" coverings, so that minimization (over the class of all schemes) does not map the class of D-schemes into itself. The primary motivation for the present paper is the question of whether the class of D-schemes has unique minimum elements. We obtain the strongest possible such theorem: for each D-scheme F there is a D-scheme $\mathbb{D}F$ which is covered by F and is such that any D-scheme strongly equivalent to F also covers $\mathbb{D}F$. Furthermore, we show that $\mathbb{D}F$ may be obtained from F by two simple types of reductions.

In order to state our main theorem, we must develop a few more definitions. It is simple to see that for any scheme F and predicate symbol π, there are coverings of $wh(\pi, 1, F)$ by $\pi(F \cdot wh(\pi, 1, F), T)$ and $wh(\pi, 2, F)$ by $\pi(T, F \cdot wh(\pi, 2, F))$. We will refer to such a morphism as a *wrap-around morphism*, or more simply as a *wrap-around*. Moreover if G and H are schemes, then there is a covering of $\pi(G, H) \cdot F$ by $\pi(G \cdot F, H \cdot F)$, which we shall refer to as a *pull-through morphism*, or simply as a *pull-through*. It should be clear that the domain of a wrap-around or pull-through is a D-scheme if and only if the same is true of its range. Let us say that a covering h of G by F has *support* $F[u, v]$ if $F[u, v]$ is a block of F and G is isomorphic to the result of substituting $h(F[u, v])$ for $F[u, v]$ in F. An *elementary reduction morphism* is a morphism with support a block on which it is a wrap-around or a pull-through. A *reduction morphism* is morphism which is a finite composition of elementary reductions. We will also say that G is a *reduction* of F is there is a reduction morphism of F onto G; if in addition, F and G both cover some scheme H, we will say that G is a *reduction over* H of F. If F covers H and there exists no reduction G of F over H with G not isomorphic to F, then we say that F is a *reduced cover* of H. We note that a reduction of a D-scheme is also a D-scheme.

We are now prepared to state the main theorem of the present paper. Recall that a D-scheme which covers a scheme H is referred to as a D-cover of H.

Theorem 5.1 *Every coaccessible, coreducible scheme H has a unique reduced D-cover $H\mathbb{D}$.*

Before proceeding to the proof of 5.1 we establish two lemmas concerning bottleneck sets in D-schemes.

Lemma 5.1 *Let S be a minimal bottleneck set for a D-scheme $G = G_1 \cdot G_2$. Then either S consists precisely of the begin of G_2 or S is a minimal bottleneck set for exactly one of G_1, G_2.*

Proof. By the minimality of S, if the begin of G_2 is a point of S it is the only point in S, so we assume that this is not the case. Let $S_i = S \cap G_i$ ($i = 1, 2$). There must be an $i \in \{1, 2\}$ such that S_i is a bottleneck set for G_i, for otherwise S is not a bottleneck set for G. Since the begin of G_2 is a bottleneck for G, every path in G is made up of a path in G_1 followed by a path in G_2 and thus must contain a point of S_i. Therefore S_i is also a bottleneck set for G. The minimality of S then implies that $S_i = S$ and the proof of the lemma is complete.

Lemma 5.2 *Let S be a minimal bottleneck set for a D-scheme G with $|S| \geq 2$. Then there is an alternation block*

$$G[y, z] = \nu(G[y_1, z], G[y_2, z])$$

of G and points s_i of S, $i = 1, 2$, such that

$$G[y_i, z] = G[y_i, s_i] \cdot G[s_i, z].$$

Proof. We proceed by induction on the number of points of G. Since the begin and exit of G are separated by S and $|S| \geq 2$, G must have at least 4 points. If G has exactly 4 points it must be the alternation of two atomic schemes and S must consist of the begins of the two atomic schemes. In this case we may set y and z equal to the begin and exit, respectively, of G to establish the conclusion.

We thus move to the inductive step. Under the hypothesis, the exit cannot be an immediate successor of the begin in G and thus G is not a while-do. If G is a nontrivial composite $G_1 \cdot G_2$ then by lemma 5.2, S is a nontrivial bottleneck set for either G_1 or G_2. In either case the inductive hypothesis holds and the conclusion follows. Thus we assume that $G = \nu(G_1, G_2)$. Let b_i denote the begin of G_i and S_i the intersection of S with G_i ($i = 1, 2$). Let i be one of 1,2. The i-outedge of b_G followed by a successful path in G_i is a successful path in G. Thus S_i is nonempty and contains a point of any given successful path in G_i; moreover S_i is minimal with respect to the latter property, so that either $S_i = b_i$ or S_i is a minimal bottleneck set for G_i. If S_i is nontrivial we may apply the inductive hypothesis to conclude the proof. Therefore, we suppose that for each $i \in \{1, 2\}$, S_i is a singleton set $\{s_i\}$ and thus that $G_i = G[b_i, s_i] \cdot G[s_i]$. Notice that the latter equation is valid regardless of whether $b_i = s_i$. The proof of the lemma is now complete as we may set $y = b_G$ and $z = e_G$.

We now turn to establishing Theorem 5.1, which states that every coaccessible, coreducible scheme H has a unique reduced D-cover $H\mathbb{D}$. The proof is by induction on the number of nodes in H, the basis step of which is trivial. Thus we henceforth fix $n > 1$, let H be a biaccessible, coreducible scheme with n points and assume the following inductive hypothesis:

> every biaccessible, coreducible scheme H' with fewer than n points has a unique reduced D-covering $H'\mathbb{D}$.

Lemma 5.3 *If G is a reduced D-cover of H and w is a bottleneck of H, then $G = G_1 \cdot G_2$ for reduced D-coverings G_1 and G_2 of $H[b_H, w]$ and $H[w]$, respectively.*

Proof. Let g be the morphism from G to H. Since every successful path of G is the b_G-lift of a successful path of H and w is a bottleneck of H, $g^{-1}(w)$ is a bottleneck set for G. Let S be a smallest minimal bottleneck set contained in $g^{-1}(w)$.

If S is nontrivial we may find a block of G of the form

$$G[y, z] = \nu(G[y_1, s_1] \cdot G[s_1, z], G[y_2, s_2] \cdot G[s_2, z])$$

as in lemma 5.3. Since z is a bottleneck rel s_i in G, $g(z)$ is a bottleneck rel $w = g(s_i)$ in H and $g(G[s_i, z]) = H[w, g(z)]$, $i = 1, 2$. Therefore $G[s_1, z]$ and $G[s_2, z]$ are both reduced covers of $H[w, g(z)]$ and hence, by the inductive hypothesis, are isomorphic. Moreover, we now see that the block $G[y, z]$ of G admits a pull-through morphism over its image under g, contradicting the assumption that G is a reduced D-covering of H. Thus we may assume that S consists of a single point u. Since u is a bottleneck of G, $G = G[b_G, u] \cdot G[u]$. Since G is a reduced D-cover of H, $G[b_G, u]$ and $G[u]$ are reduced covers of $H[b_H, w]$ and $H[w]$, respectively, so the proof of the lemma is complete.

Lemma 5.4 *If H is a while-do of some scheme H_1, then the while-do of $H_1\mathbf{D}$ is the unique reduced D-cover of H.*

Proof. Without loss of generality we assume $H = wh(\pi, 1, H_1)$. If G' is the corresponding while-do of $H_1\mathbf{D}$, then clearly G' is a reduced D-cover of H. We now prove uniqueness by showing that any reduced D-cover of H is isomorphic to G'. Let G be a reduced D-cover of H. Then, since H has no bottlenecks, G is not a nontrivial composition. If G is a while-do, then it is the while-do of a reduced D-cover of H_1 and hence is isomorphic to G', as claimed.

Suppose now that there is a reduced D-cover of H distinct from G' and let G be a smallest such scheme. By the above, G must be an alternation and thus, since the 2-successor of b_H is the exit of H, $G = \pi(F, T)$ for some scheme F. If x is the 1-successor of b_H — i.e., the begin of H_1 – then F must be a reduced D-cover of the coreducible scheme $H' = H[x]$ whose begin is x. Since the begin of H is a bottleneck rel x in H, it is a bottleneck of H'. By lemma 5.4, $F = F_1 \cdot F_2$, where F_1 is a reduced D-covering of $H'[x, b_H] = H[x, b_H] = H_1$ and F_2 is a reduced D-covering of $H'[b_H] = H$. By the inductive hypothesis, F_1 is isomorphic to $H_1\mathbf{D}$. Now F_2 has fewer points than G and is a reduced cover of H; since G was the smallest reduced cover of H distinct from G', we conclude that F_2 is isomorphic to $G' = wh(\pi, 1, H_1\mathbf{D})$. But then G admits a pull-through over H. This contradiction completes the proof of lemma 5.5.

Lemma 5.5 *If H has a bottleneck, then there is a unique reduced D-cover of H.*

Proof. It is simple to see that if w is a bottleneck of H, then $H[b_H, w] \cdot H[w]$ covers H. Since $H[b_H, w]$ has fewer points than H, it has a unique reduced D-cover G_1, and thus $G_1 \cdot H[w]$ covers H. If b_H is not in $H[w]$, then $H[w]$ also has a unique reduced D-cover G_2, and thus $G = G_1 \cdot G_2$ is a D-cover of H. Since the support of an elementary reduction of a composition $G_1 \cdot G_2$ must be contained in one of G_1, G_2, G is also reduced. Uniqueness then follows directly from lemma 5.4.

We thus assume that b_H is accessible from every bottleneck of H. Let w be a bottleneck of H. Then b_H is in $H[w]$, from which it is obvious that w must be the exit point for the strong component K of b_H. Let v be the immediate successor of w not in K. Since every successful path must contain v and b_H is not accessible from v, v must be the exit of H. But then $H[w]$ is a while-do and thus, by lemma 5.6, has a unique reduced D-cover G_2. As shown in the previous paragraph, $G = G_1 \cdot G_2$ is then the unique reduced D-cover of H, and the proof of lemma 5.6 is complete.

We are now prepared to complete the proof of theorem 5.1 by establishing the following.

Lemma 5.6 *If H has no bottlenecks and is not a while-do, then H has a unique reduced D-cover.*

Proof. If b_H has nontrivial strong component K, then K has a unique exit u. Since H has no bottlenecks, $u = b_H$ and the successor of b_H not in K must be the exit of H. Since in this latter case H would be a while-do, we may conclude that b_H has trivial strong component and hence has indegree zero. Moreover, b_H may not be labeled by an operator symbol since that would imply that H is a nontrivial composition and hence has a bottleneck.

We thus have that b_H has no in-edges and is labeled by some predicate symbol π. It then follows that no while-do scheme can cover H. Since H has no bottlenecks, no nontrivial composition can cover H. Thus every D-cover of H is an alternation. For $i = 1, 2$, let x_i be the i-successor of b_H in H; then b_H is not in $H[x_i]$ and thus, by the inductive hypothesis, there is a unique reduced D-cover G_i of $H[x_i]$. Letting $G = \pi(G_1, G_2)$, we see that G is a D-cover of H which is either reduced or admits a global reduction over H. But the latter is not possible since H may not be covered by either a nontrivial composition or a while-do. Uniqueness is now a simple matter since any reduced covering of H must be the alternation of reduced coverings of $H[x_1]$ and $H[x_2]$ and thus is isomorphic to G, the alternation of their unique reduced D-covers.

6 Corollaries to the main theorem

Theorem 5.1 is a condensation of many distinct results into one compact statement, which we now unravel as corollaries. Since we did not assume the converse of proposition 4.6, it follows as the following corollary of 5.1.

Corollary 6.1 *Every biaccessible, coreducible scheme is covered by a D-scheme.*

Since a D-scheme is its own unique reduced D-cover the next result is immediate.

Corollary 6.2 *Every morphism between D-schemes is a composition of morphisms each of which has support a block on which it is a wrap-around or a pull-through.*

We remark that Douglas Troeger has used corollary 6.2 as the basis for an axiomatization of the algebra of strongly equivalence classes of D-schemes[DT]. The next corollary could be viewed as the "unique minimum D-schemes" theorem referred to earlier in the paper. For any D-scheme F, let $\mathbf{D}(F)$ be the unique reduced D-cover of the minimization $F\mathbf{M}$ of F over the class of all schemes.

Corollary 6.3 *Let F and G be D-schemes. Then*

a) F is strongly equivalent to $\mathbf{D}(F)$;

b) $\mathbf{D}(F)$ has the fewest nodes among all D-schemes strong equivalent to F;

c) G is strongly equivalent to F if and only if $\mathbf{D}(F)$ is isomorphic to $\mathbf{D}(G)$; and

d) if G is strongly equivalent to F and has the same number of nodes as $\mathbf{D}(F)$, then G is isomorphic to $\mathbf{D}(F)$.

References

[BT] S. L. BLOOM AND R. TINDELL, Algebraic and graph-theoretic characterizations of structured flowchart schemes, *Theoret. Comput. Sci.* **9** (1979), 256-286.

[CE] C. C. ELGOT, Structured programming with and without GO TO statements, *IEEE Trans. Software Engrg.* **SE-2**, 41-53; erratum and corrigendum, p. 232.

[DT] D. R. TROEGER, An axiomatization of D-scheme strong equivalence, *J. Comp. Sys. Sci.* **27** (1983), 221-224.

(Received November 12, 1990)

Acta Cybernetica, Tom. 10, Fasc. 1-2, Szeged, 1991

A note on fully initial grammars

S. Vicolov*

We (negatively) solve two conjectures of Mateescu and Paun [3], then we give characterizations in terms of syntactic semigroup of some families of regular fully initial languages.

1 Definitions and notations

For a vocabulary V, we denote by $V^*(V^+)$ the free monoid (semigroup) generated by V under the operation of concatenation; λ is the null element $(V^+ = V^* - \{\lambda\})$. The strings of V^* are called words. The length of a word $x \in V^*$ is denoted by $|x|$.

If we consider a Chomsky grammar $G = (V_N, V_T, S, P)$, then the usual language generated by G is defined by

$$L(G) = \{x \in V_T^* | S \overset{*}{\Longrightarrow} x\}.$$

The fully initial language generated by G is

$$L_{in}(G) = \{x \in V_T^* | A \overset{*}{\Longrightarrow} x \text{ for some } A \in V_N\}.$$

The study of fully initial languages was proposed by S. Horvath and has been done in a series of papers [1], [2], [3], [4].

Clearly, $L(G) \subseteq L_{in}(G)$. The family of fully initial languages generated by grammars of type $i, i = 0, 1, 2, 3$ is denoted by $\mathcal{FL}_i$.

Usually, the right-linear and the left-linear grammars generate the same family of languages. For fully initial grammars this is not true, therefore we shall distinguish several classes of "type-3" grammars.

A grammar $G = (V_N, V_T, S, P)$ is called right-linear (left-linear) if $P \subseteq V_N \times (V_T^* \cup V_T^* V_N)(P \subseteq V_N \times (V_T^* \cup V_N V_T^*))$. We denote by $\mathcal{FL}_{rlin}, \mathcal{FL}_{llin}$ the corresponding families of fully initial languages. A grammar $G = (V_N, V_T, S, P)$ is called right-regular (left-regular) if $P \subseteq V_N \times (V_T \cup V_T V_N)(P \subseteq V_N \times (V_T \cup V_N V_T))$. The corresponding families of fully initial languages are denoted by $\mathcal{FL}_{rreg}, \mathcal{FL}_{lreg}$.$\mathcal{FL}_3$ is, in fact, $\mathcal{FL}_{rlin} \cup \mathcal{FL}_{llin}$. Following [3] we shall consider the next families, too:

$$\mathcal{FL}_{reg}^{\cap} = \mathcal{FL}_{rreg} \cap \mathcal{FL}_{lreg}$$

$$\mathcal{FL}_{reg}^{\cup} = \mathcal{FL}_{rreg} \cup \mathcal{FL}_{lreg}.$$

The sets of prefixes, suffixes and subwords of a given word x are denoted by $\mathrm{Init}(x)$, $\mathrm{Fin}(x)$, $\mathrm{Sub}(x)$, respectively, and these notations will be extended in the

*University of Bucharest, Faculty of Mathematics, Str. Academiei 14,70109 Bucuresti ROMANIA

natural way to languages. When considering only proper prefixes, suffixes and subwords, we shall write $\text{Initp}(x)$, $\text{Finp}(x)$ and $\text{Subp}(x)$, respectively.

Let L be a language of V^+. The congruence $\sim_L$ defined over V^+ by: $u \sim_L v$ if and only if, for every $x, y \in V^*, xuy \in L \Leftrightarrow xvy \in L$, is called the syntactic congruence of L. The syntactic semigroup of L is the quotient semigroup $A^+/\sim_L$.

For further details in syntactic semigroup theory, the reader is referred to [5].

2 Necessary conditions for the context- free case

We shall reproduce here the necessary conditions for a language to be in $\mathcal{FL}_2$, which were considered in [3]. Finally we shall prove that two of the conjectures formulated there are not true.

Lemma 1 *For each language $L \in \mathcal{FL}_2, L \subseteq V^*$, there are two positive integers p, q such that each $z \in L, |z| > p$, can be written as $z = uvwxy, u, v, w, x, y \in V^*$, so that*

(i) $|vwx| \leq q, \quad |vx| > 0$,

(ii) for all $k \geq 0, uv^k wx^k y \in L$ and $v^k wx^k \in L$.

Definition 1 *For a given language $L \subseteq V^*$, let*

$$\text{Min}(L) = \{z \in L | \text{Subp}(z) \cap L = \emptyset\}$$

and define

$$R_1(L) = \text{Min}(L)$$

$$R_i(L) = R_{i-1}(L) \cup \text{Min}(L - R_{i-1}(L)), i \geq 2.$$

We say that L has property R if and only if all the sets $R_i(L), i \geq 1$, are finite.

Lemma 2 *If $L \in \mathcal{FL}_2$, then L has property R.*

In [3] it is also proved that none of these conditions is sufficient for a language to be in $\mathcal{FL}_2$, and one formulates the following conjectures:

(1) If L is a context-free language which fulfils the condition in Lemma 1, then $L \in \mathcal{FL}_2$.

(2) For arbitrary languages, the condition in Lemma 1 is stronger than property R.

Proposition 1 *Conjecture (2) is not true.*

Proof. Consider the languages

$$L_1 = \{cd^n ae^{k_1}b \ldots e^{k_n}b | n \geq 0, k_1, \ldots k_n \geq 0\},$$

$$L = L_1 \cup \{e^n b | n \geq 0\} \cup \{d^n ab^n | n \geq 0\}.$$

We shall prove that L fulfils the condition in Lemma 1. Let us take $p = 2$ and $q = 3$. For $z = e^n b$ or $z = d^n ab^n$ we clearly have all conditions in lemma fulfilled. If $z = cd^n ae^{k_1}b \ldots e^{k_n}b$, then $|z| > p$ implies $n \geq 1$. There are two cases.

1. For all $i, 1 \le i \le n, k_i = 0$. Therefore $z = cd^n ab^n$. We take $u = cd^{n-1}, v = d, w = a, x = b, y = b^{n-1}$. It follows that $z = uvwxy, |vx| > 0, |vwx| \le q, uv^k wx^k y = cd^{n-1}d^k ab^k b^{n-1} \in L$ and $v^k wx^k = d^k ab^k \in L$ for every $k \ge 0$.

2. There is an $i, 1 \le i \le n$, such that $k_i \ge 1$. We consider $u = cd^n ae^{k_1}b \dots e^{k_{i-1}}be^{k_i - 1}, v = e, w = b, x = \lambda, y = e^{k_{i+1}}b \dots e^{k_n}b$. Then $z = uvwxy, |vx| > 0, |vwx| \le q, uv^k wx^k y = cd^n ae^{k_1}b \dots e^{k_{i-1}}be^{k_i-1}e^k be^{k_{i+1}}b \dots e^{k_n}b \in L$ and $v^k wx^k = e^k b \in L$ for all $k \ge 0$.

On the other hand, L does not observe property R. Indeed, it is clear that $R_1(L) = \{a, b\}$ and $R_2(L) = \{a, b, ca, eb, dab\}$. $\mathrm{Min}(L - R_2(L)) \supseteq \{cd^n a(eb)^n | n \ge 1\}$ since, for all $n \ge 1, z = cd^n a(eb)^n$ implies $z \in L - R_2(L), \mathrm{Subp}(z) \cap L_1 = \emptyset$ and $\mathrm{Subp}(z) \cap (L - L_1) = \{a, b, eb\} \subseteq R_2(L)$. It follows that $R_3(L)$ is an infinite set.

In conclusion, L fulfils the condition in Lemma 1 without observing property R.

Proposition 2 *Conjecture (1) is not true.*

Proof. We shall consider the same language L as in the above proof. Let $G = (V_N, V_T, S, P)$, where $V_N = \{A, B, C, S\}, V_T = \{a, b, c, d, e\}$ and $P = \{S \longrightarrow cA, A \longrightarrow dAB, B \longrightarrow eB, A \longrightarrow a, B \longrightarrow b, S \longrightarrow B, S \longrightarrow C, C \longrightarrow dCb, C \longrightarrow a\}$. It is easy to see that $L = L(G)$. Consequently, L is a context - free language which fulfils the condition in Lemma 1. L has not property R, therefore, according to Lemma 2, $L \notin \mathcal{FL}_2$. In conclusion, the proposition is proved.

Remark 1 *Note that* $L_{in}(G) = L \cup \{d^n ae^{k_1}b \dots e^{k_n}b | n \ge 0, k_i \ge 0, 1 \le i \le n\}$.

Remark 2 *The negative answer of these two conjectures raises another problem: a context-free language which satisfies simultaneously the condition in Lemma 1 and the condition R, is in* $\mathcal{FL}_2$*?*

Proposition 3 *The condition R and the condition in Lemma 1 fulfilled in the same time, are not sufficient for a context-free language to be in* $\mathcal{FL}_2$.

Proof. Consider the language

$$L_2 = \{cd^n ae^{k_1}b \dots e^{k_n}b | n \ge 0, k_1, \dots k_n \ge 0\} \cup \{d^n ab^n | n \ge 0\} \cup \{e, b\}^+.$$

Note that $L_2 = L \cup \{e, b\}^+$, where L is the language used in the above proofs. L and $\{e, b\}^+$ are context-free languages. Consequently, L_2 is a context-free language, too. We have pointed out in the proof of Proposition 1 that L satisfies the condition in Lemma 1; it is easy to see that $\{e, b\}^+$ also satisfies this condition. In conclusion, L_2 fulfils the condition in Lemma 1.

L_2 observes property R. Indeed, $R_1(L_2) = \{a, e, b\}$ and $R_i(L_2) = \{cd^n ae^{k_1}b \dots e^{k_n}b | 0 \le n \le i - 2, 0 \le n + k_1 + \dots + k_n \le i - 1\} \cup \{d^n ab^n | 0 \le n \le i - 1\} \cup \{u \in \{e, b\}^+, |u| \le i\}, i \ge 2$.

The last equality can be obtained by induction. We denote by A_i the right term of the equality. It is clear that $R_2(L_2) = A_2$. Suppose that $R_j(L_2) = A_j$, for an arbitrary $j \ge 2$. We must show that $R_{j+1}(L_2) = A_{j+1}$. According to definition and to the above supposition we have $R_{j+1}(L_2) = R_j(L_2) \cup \mathrm{Min}(L_2 - R_j(L_2)) = A_j \cup \mathrm{Min}(L_2 - A_j)$. Also using the inclusions $A_{j+1} \subseteq L_2$ and $R_{j+1}(L_2) \subseteq L_2$, we conclude that it is sufficient to prove that $z \in A_{j+1}$ iff $z \in A_j \cup \mathrm{Min}(L_2 - A_j)$, for all $z \in L_2$. There are three cases.

(1) $z = cd^n ae^{k_1}b \dots e^{k_n}b$. $z \in A_{j+1}$ if $n \leq j-1$ and $n + k_1 + \dots + k_n \leq j$. Obviously, $\mathrm{Subp}(z) \cap L_2 = \mathrm{Sub}(e^{k_1}b \dots e^{k_n}b) \cup \{d^t ab^t | 1 \leq n, k_1 + \dots + k_t = 0\}$.

Suppose that $z \in A_{j+1}$. We obtain $\mathrm{Subp}(z) \cap L_2 \subseteq \{u \in \{e,b\}^+ || u| \leq j\} \cup \{d^t ab^t | t \leq j-1\} \subseteq A_j$. It follows that $z \in A_j \cup \mathrm{Min}(L_2 - A_j)$.

Conversely, suppose that $z \in A_j \cup \mathrm{Min}(L_2 - A_j)$. If $z \in A_j$, then $z \in A_{j+1}$. If $z \in \mathrm{Min}(L_2 - A_j)$, we obtain $\mathrm{Subp}(z) \cap L_2 \subseteq A_j$. This implies $\mathrm{Sub}(e^{k_1}b \dots e^{k_n}b) \subseteq A_j$. Hence $n + k_1 + \dots + k_n \leq j$ and $n \leq j$. If $n = j$, we have $k_1 + \dots + k_n = 0$ and $d^j ab^j \in (\mathrm{Subp}(z) \cap L_2) - A_j$, which is a contradiction. Consequently, $n \leq j-1$ and $n + k_1 + \dots + k_n \leq j$.

Thus we proved that, in this case, $z \in A_{j+1}$ iff $z \in R_{j+1}(L_2)$.

(2) $z = d^n ab^n$. $z \in A_{j+1}$ iff $n \leq j$. $n \leq j$ iff $\mathrm{Subp}(z) \cap L_2 = \{d^k ab^k | k \leq j-1\} (\subseteq A_j)$ iff $z \in A_j \cup \mathrm{Min}(L_2 - A_j)$.

(3) $z \in \{e,b\}^+$. $z \in A_{j+1}$ iff $|z| \leq j+1$ iff $\mathrm{Subp}(z) \cap L_2 \subseteq \{u \in \{e,b\}^+ | \ |u| \leq j\} (\subseteq A_j)$ iff $z \in A_j \cup \mathrm{Min}(L_2 - A_j)$.

In conclusion, L_2 is a context-free language which satisfies both the condition in Lemma 1 and the condition R.

On the other hand, $L_2 \notin \mathcal{FL}_2$. Assume the contrary and consider a type-2 grammar $G = (V_N, V_T, S, P)$ such that $L_{in}(G) = L_2$. Since $L_2 = \{cd^n ae^{k_1}b \dots e^{k_n}b | n \geq 0, k_1, \dots, k_n \geq 0\} \cup \{d^n ab^n | n \geq 0\} \cup \{e,b\}^+$, we conclude that, for generating the strings of the form $cd^n ae^{k_1}b \dots e^{k_n}b$, we need derivations such as: $X \overset{*}{\Longrightarrow} d^j X B^j, j \geq 1, X \in V_N, B \in V_N, B \overset{*}{\Longrightarrow} e^k b, k \geq 1, X \overset{*}{\Longrightarrow} w, w \in T_T^+$. It follows that $d^j w (e^k b)^j \in L_{in}(G) - L_2$, which is a contradiction.

Thus, the proof is completed.

3 Characterizations of languages in $\mathcal{FL}_{rreg}$, $\mathcal{FL}_{lreg}$, $\mathcal{FL}_{reg}^{\cap}$

We shall consider here a characterization of these families in terms of the syntactic semigroup. For proving it we shall use the following lemma, presented in [3].

Lemma 3 *(i) $L \in \mathcal{FL}_{rreg}$ if and only if L is regular and $L = Fin(L)$.*
(ii) $L \in \mathcal{FL}_{lreg}$ if and only if L is regular and $L = Init(L)$.
(iii) $L \in \mathcal{FL}_{reg}^{\cap}$ if and only if L is regular and $L = Sub(L)$.

We also shall use two well-known results in the theory of syntactic semigroups [5]:

Lemma 4 *Let $L \subseteq V^+$. L is regular if and only if its syntactic semigroup is finite.*

Lemma 5 *Let $L \subseteq V^+$ be a language and denote by φ the canonical homomorphism φ: $V^+ \longrightarrow V^+/\sim_L$. Then $V^+ - L = \varphi^{-1}(\varphi(V^+ - L))$.*

We shall consider below that L, $\mathrm{Fin}(L)$, $\mathrm{Init}(L)$ and $\mathrm{Sub}(L)$ do not contain the null word λ.

Proposition 4 *Let L be a language over V. Denote by S the syntactic semigroup of L, by φ the canonical homomorphism $\varphi : V^+ \longrightarrow V^+/\sim_L = S$ and $P = \varphi(L)$. Then, we have:*

(i) $L \in \mathcal{FL}_{rreg}$ if and only if S is finite and $S(S-P) \subseteq S-P$.

(ii) $L \in \mathcal{FL}_{lreg}$ if and only if S is finite and $(S-P)S \subseteq S-P$.

(iii) $L \in \mathcal{FL}^{\cap}_{reg}$ if and only if S is finite, S has a zero, 0, and $S-P=\{0\}$.

Proof. (i) According to Lemma 3, part (i), $L \in \mathcal{FL}_{rreg}$ if and only if L is regular and $L = \mathrm{Fin}(L)$. Since we always have $L \subseteq \mathrm{Fin}(L)$, we deduce that $L = \mathrm{Fin}(L)$ is equivalent to "for all $u, v \in V^+, uv \in L \Longrightarrow v \in L$", statement which is also equivalent to "for all $u \in V^+$ and $v \in V^+ - L, uv \in V^+ - L$", i.e. $V^+(V^+-L) \subseteq V^+ - L$. It follows from the last inclusion that $\varphi(V^+(V^+-L)) \subseteq \varphi(V^+-L)$ and hence $\varphi^{-1}(\varphi(V^+(V^+-L))) \subseteq \varphi^{-1}(\varphi(V^+-L))$. In turn, the last inclusion implies $V^+(V^+-L) \subseteq V^+-L$, since $V^+(V^+-L) \subseteq \varphi^{-1}(\varphi(V^+(V^+-L)))$ and $\varphi^{-1}(\varphi(V^+-L)) = V^+-L$ (Lemma 5). Consequently, $V^+(V^+-L) \subseteq V^+-L$ if and only if $\varphi(V^+)\varphi(V^+-L) \subseteq \varphi(V^+-L)(\varphi(V^+(V^+-L)) = \varphi(V^+)\varphi(V^+-L)$ since φ is homomorphism of semigroups) if and only if $S(S-P) \subseteq S-P$ (use $\varphi(V^+) = S$ and $\varphi(V^+-L) = S-P$, from Lemma 5). Thus we proved the equivalence between $L = \mathrm{Fin}(L)$ and $S(S-P) \subseteq S-P$. Using the result in Lemma 4, too, we conclude the proof.

(ii) The proof is symmetrical.

(iii) Suppose that $L \in \mathcal{FL}^{\cap}_{reg}$. According to Lemma 3, part (iii), L is regular and $\mathrm{Sub}(L) = L$. From the last equality it follows that "$u \notin L \Longrightarrow xuy \notin L$, for all $x, y \in V^*$ and $u \in V^+$" (assuming the contrary, we have $xuy \in L$, hence $u \in \mathrm{Sub}(L) = L$, which is a contradiction to $u \notin L$). Take u, v arbitrary in V^+ such that $u \notin L$. From the above statement we obtain $uv \notin L, vu \notin L$ and: "$xuy \notin L, xuvy \notin L, xvuy \notin L$, for every $x, y \in V^+$". Consequently $u \sim_L uv \sim_L vu$ and hence we have $\varphi(u) = \varphi(uv) = \varphi(vu)$, i.e. $\varphi(u) = \varphi(u)\varphi(v) = \varphi(v)\varphi(u)$. Since v is an arbitrary word of $V^+, \varphi(v)$ is an arbitrary element of $\varphi(V^+) = S$. Therefore we deduce that $\varphi(u)$ is a zero of S. A semigroup may contain only one zero. As u is arbitrary in V^+-L and $\varphi(V^+-L) = S-P$, we conclude that $S-P$ contains only one element, which is the zero of S. Since L is regular, S is finite. Thus, one of the implications is proved.

Conversely, suppose that S is finite, S has a zero, 0, and $S-P=\{0\}$. Clearly, $(S-P)S \subseteq S-P$ and $S(S-P) \subseteq S-P$. According to the parts (i) and (ii) of this Proposition, it follows that $L \in \mathcal{FL}^{\cap}_{reg}$.

Corollary 1 *Let L be a language of V^+ whose syntactic semigroup is commutative. If $L \in \mathcal{FL}^{\cup}_{reg}$, then in fact L is in $\mathcal{FL}^{\cap}_{reg}$.*

Proof. $L \in \mathcal{FL}^{\cup}_{reg}$ implies $L \in \mathcal{FL}_{rreg}$ of $L \in \mathcal{FL}_{lreg}$. We use Proposition 4, parts (i), (ii), and we obtain $S(S-P) \subseteq S-P$ or $(S-P)S \subseteq S-P$. Since S is commutative, these inclusions hold simultaneously. Using again Proposition 4, parts (i), (ii), we conclude that $L \in \mathcal{FL}^{\cap}_{reg}$.

References

[1] T. Balanescu, M. Gheorghe, Gh. Paun, On fully initial grammars with regulated rewriting, Acta Cybernetica, 9, 2(1989), 157-165.

[2] J. Dassow, On fully initial context-free languages, Papers on Automata and Languages (Ed. I. Peák), X(1988), 3-6.

[3] A. Mateescu, Gh. Paun, Further remarks on fully initial grammars, Acta Cybernetica, 9, 2 (1989), 143-156.

[4] Gh. Paun, A note on fully initial context-free languages, Papers on Automata and Languages, X (1988), 7-11.

[5] J.E. Pin, Varieties of formal languages, North Oxford Academic Publishers, London, 1986, 5-24.

(Received July 7, 1990)

Készítette a JATEPress
6722 Szeged, Petőfi S. sgt. 30 – 34.

Subscription information and mailing address for editorial correspondence:

Acta Cybernetica
Árpád tér 2.
Szeged
H-6720 Hungary

INDEX — TARTALOM

ISSN 0324—721 X

Felelős szerkesztő és kiadó: Gécseg Ferenc
A kézirat a nyomdába érkezett: 1991. okt.
Terjedelem: 11,5 (A/5) ív